Vol. 43 No. 2 IPSJ Journal Feb. 2002

Regular Paper

Facilitator Agent Based on Word-of-mouth Trust

Chihiro Ono,† Dai Kanetomo,†† Keesoo Kim,†††
Boyd C. Paulson, Jr.,††† Mark Cutkosky†††
and Charles J. Petrie, Jr.†††

In this paper, we propose a facilitator which finds capable and trustworthy partners on behalf of client users, which helps users form and maintain e-partnerships for electronic commerce and electronic collaboration. Unlike existing capability-based facilitators or matchmakers, the facilitator collects and maintains private "word-of-mouth" trust information as well as capabilities from each user and uses the information for personalized trust-based facilitation for each user, which is done through the facilitation protocols and trust propagation mechanism. Compared to other existing trust mechanisms, the characteristics of trust which this facilitator handles are personalized-collaborative-subjective-qualitative-private. The facilitator was implemented as a JATLite multi-agent system and tested in the area of construction supply-chain coordination.

1. Introduction

Currently, online communities where electronic commerce and electronic collaboration are carried out are rapidly expanding along with the growth of the Internet. In these communities, there may be negotiation among automated software programs, called agents. For instance, at auction services, many sellers create auctions for various kinds of goods and many potential buyers are bidding for goods by following auction protocols. In construction projects, subcontractors negotiate schedules and tasks with general contractors.

In on-line communities for electronic commerce and electronic collaboration, establishing partnerships with which participants (agents) can interact or trade with each other, which we call e-partnerships, is crucial to many applications, such as online auctions and project coordination in various industries. In these cases, agents must have a mechanism for establishing and maintaining partnerships of personally trusted agents, which is based on private word-of-mouth trust information. Also, the partnerships must be dynamic and able to be formed rapidly as application needs dictate, and agents must be able to join the partnerships or be rejected as appropriate.

Our understanding and assumptions of online communities are: 1) There are many participant agents and most of them do not know each other. 2) Agents join or leave the community very often. 3) Agents want to keep their opinions of other agents secret. Under those assumptions, it is very important to dynamically find a group of appropriate partners to negotiate with out of a large number of potential partners. This is because, at auction services for example, sellers or auctioneers have to notify potential buyers about the creation of new auctions or, in construction projects, subcontractors have to find potential partners with them to negotiate tasks or schedules.

So far, for the task of finding partners, a concept of facilitator and matchmaker has been proposed. **Figure 1** shows a general form of facilitation [12].

First, provider agents advertise their capabilities to a facilitator agent and the facilitator stores these advertisements. When a requester asks a facilitator whether it knows of providers with the desired capabilities, the facilitator matches the request against the stored advertisements and returns the result, a subset of stored advertisements.

There are several standards which have facilitator-like servers for making dynamic e-partnerships. The Knowledge Query and Manipulation Language (KQML) [11], proposed as a standard for an agent communication language, assumes the existence of a facilitator and several protocols for facilitation are defined: broker, recommend, recruit, and subscribe. The Foundation for Intelligent Physical Agents (FIPA) [7], a standardization body

† KDDI R&D Laboratories Inc.
†† NTT Comware Co.
††† Stanford University

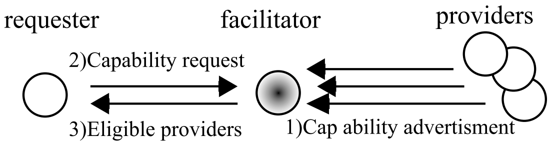


Fig. 1 General mechanism of facilitator.

for agent-related technologies such as an agent communication language and agent management, also has a facilitator called Directory Facilitator (DF). The Common Object Request Broker Architecture (CORBA)[4], a standard for developing large-scale distributed object-oriented applications, also has a facilitation server called TRADER. Jini[8], an architecture for developing Java-based distributed applications proposed by SUN Microsystems, also has a facilitation server called Lookup Server. In addition to these standards, there also exist some implementations, such as Matchmaker by CMU[12] and Kasbah by MIT[3].

Unfortunately, however, all of them perform facilitation based only on the registered capabilities of service provider agents and are not sufficient for making e-partnerships under our assumptions. This is because requesters do not want to deal with bad providers. On the other hand, providers also do not want to deal with bad requesters. It is then necessary to filter and rank requests and responses according to trustworthiness for both requesters and providers.

When we think of trust information in the "real world," word-of-mouth information is considered to be very important. There are many quotations on the value of word of mouth: "The best prospect is the client who has already dealt with you. The second best is the one referred to by a client who has dealt with you previously. The third best is the one referred to you by another trusted professional or friend" (Marilyn Jennings)[2]. "Forget about market surveys and analyst reports. Word of mouth is probably the most powerful form of communication in the business world. It can either hurt a company's reputation or ..." (Regis McKenna and others)[2]. Considering this, using word-of-mouth, private trust information seems to be better than using third-party rating systems such as market surveys.

Thus, this paper proposes a private trust-based facilitator for forming e-partnerships which find partners based on trustworthiness as well as the capabilities of service provider agents. Section 2 describes the categorization of trust and the trust model. Section 3 explains the design of the facilitator including protocols and the inside mechanism. Section 4 shows the implementation and Section 5 gives an example of using the facilitator. Finally, we conclude the paper and discuss future work in Section 6.

2. Trust-based Facilitation

"Trust" information, which we try to make use of in facilitation, has been defined and used differently in many applications and services such as rating systems and reputation systems. In this section, we categorize characteristics of trust.

2.1 Trust for e-Partnerships

Roughly, we define trust as a general factor for deciding whether or not the facilitator can introduce the agents, as is shown in Zolin's definition[14]: "Trust is the deciding factor in a social process that results in a decision by an individual to accept or reject a risk based on the expectation that another party will perform to the individual's expected performance requirements (p.875)." And we call trust the value trustworthiness. As this definition is too vague, however, we define five characteristics of trustworthiness:

(1) Commonality of trustworthiness of target agent
Standardized: same for all participant agents
Personalized: different from each other
(2) Evaluator of target agent
Authoritative: third-party authority
Collaborative: participants
(3) Objectivity of evaluation
Objective: based on common criteria
Subjective: based on different criteria
(4) Complexity of trustworthiness
Quantitative: numeric values
Qualitative: boolean (positive or negative)
(5) Disclosures of reputation report
Public: open to public
Private: closed to public

Based on this characterization, existing applications are categorized as shown in **Fig. 2**.

The Better Business Bureau (BBB)[1] and Dun & Bradstreet (D&B)[5] rate companies and provide information to those who inquire about inquired companies, which include trustworthiness, management, profit and so on. BBB has more than 8,000 member companies and D&B has rating information for 58 million compa-

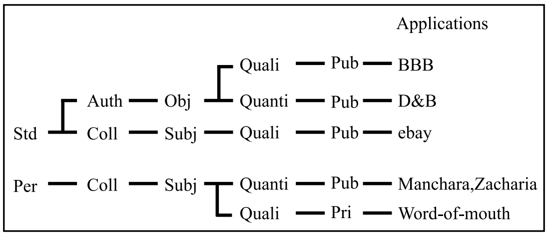


Fig. 2 Categorization of applications.

nies. Both the BBB and D&B evaluate companies by themselves (Authoritative) based on a certain criteria (Objective) and provide a common trustworthiness of target agents (Standardized). For the complexity of trustworthiness, BBB provides a Boolean rating. On the other hand, D&B provides numeric ratings of companies.

eBay[6] runs an auction site to sell and buy various goods, which has more than 10 million members. On eBay, sellers and buyers can check rating scores of potential partners before trading as they evaluate each other by providing feedback after their trades, which means eBay belongs to Collaborative and Subjective. Also, for rating, it belongs to Standardized, Qualitative and Public.

There are several algorithms for handling Personalized Distributed Subjective trustworthiness. Manchala, et al.[10] propose trust metrics and models for e-commerce by calculating over a chain of numerical trust values (Quantitative) when there is a public intermediary. However, this method is not sufficient for use in trust-based facilitation because the method of building a chain of agents is not mentioned and calculation of numeric values is too complicated. Zacharia, et al.[13] propose a collaborative reputation mechanism between source and target agents. However, this method also is not sufficient as the calculation used here is overly complex, especially for the calculation of numeric values (Quantitative) and all the paths, including those unused.

2.2 Handling Word-of-mouth Trust

For taking advantage of word-of-mouth trust information and existing facilitators, which collect capabilities registered by provider agents and do facilitation based on that information, we propose a facilitator which collects trust information from participants as well as capabilities, and uses both of them for facilitation.

Requirements for word-of-mouth trust-based facilitation are as follows:

(1) As for commonality, we choose "Personalized". That is, trustworthiness of target agent (agent_A) for one source agent (agent_B) is different from that for another source agent (agent_E). Therefore, a facilitator should keep as many as $n(n-1)$ patterns of trustworthiness.

(2) As for the evaluator, we choose "Collaborative."

(3) As for objectivity, we choose "Subjective".

(4) As for the complexity, we choose "Qualitative" as the calculation should be simple enough to update $n(n-1)$ patterns of trustworthiness.

(5) As for disclosures of reputation, we choose "Private" as we assume that participants want to keep their opinions of other agents secret.

(6) Trustworthiness should be transitive. That is, if agentA directly trusts agentB and agentB directly trusts agentC, agentA can indirectly trust agentC.

(7) Trustworthiness should be kept different by capabilities. That is, the trustworthiness of one agent concerning car sales could be different from that of the same agent concerning car repair.

In summary, the characteristics of trustworthiness which a facilitator handles is private-collaborative-subjective-qualitative-private.

2.3 Representation of Trustworthiness

The way of representing trustworthiness has the following types based on the requirements described in Section 2.2. First, cases in which an agent evaluates the target agent directly based on its own previous experience with requested capabilities include:

- Direct positive reputation (DP): A source agent trusts a target agent directly.
- Direct negative reputation (DN): A source agent distrusts a target agent directly.

Second, cases in which an agent evaluates the target agent by using chain of trustworthiness from the source agent to the target agent include:

- Indirect positive reputation (IP): A source agent trusts a target agent indirectly.
- Indirect negative reputation (IN): A source agent distrusts a target agent indirectly.

Finally, the case in which an agent has no information about the target agent includes:

- Unknown (UN): A source agent cannot decide whether it can trust or distrust a tar-

Trust Table for capability_X						
	AA	AB	AC	AD	AE	AF
Agent A		DP	IP	IP	DP	UN
Agent B	DP		DP	UN	DP	UN
Agent C	IN	DN		DN	DP	UN
Agent D	IP	DP	IP		IP	DP
Agent E	DN	IP	DP	DP		UN
Agent F	UN	IP	DP	DP	UN	

Fig. 3 Example of trust table.

get agent.

Thus, trustworthiness can be represented by any of five types and they are kept in n^*n table for each capability as shown in **Fig. 3**. In this example, an agentA directly trusts agentE, but agentE distrusts agentA.

3. Facilitator Design

Based on the policy discussed above, a facilitator collects capabilities and trust information from participant agents and maintains this information for each capability and uses them in facilitation. There is an approach other than having a facilitator for finding partners, distributed way, with which each participant agent keeps its information by itself and exchanges it with each other. However, using facilitator is better because keeping opinions of other agents secret and maintaining a large amount of capability/trust information in distributed way are very complicated. This section describes a protocol for using a facilitator, registration and request method, and message format and maintenance method of trustworthiness.

3.1 Facilitation Protocols

For communication among requesters, providers and a facilitator, we use KQML[11], which provides protocols for facilitation: broker, recruit, recommend, and subscribe for requests, as shown in **Fig. 4**.

Both capabilities and trustworthiness are described in the "content" parameter of KQML. **Figure 5** shows a message format of the content parameter. A "Message" comprises message type, capability information and trust information. A message type includes REGISTRATION, QUESTION and ANSWER. Capability information includes context type, condition, extract terms and result. "Context type" is a name of capability, and we assume global namespaces, in which all the participants have a common vocabulary about their capabilities and attributes. From them, only the context

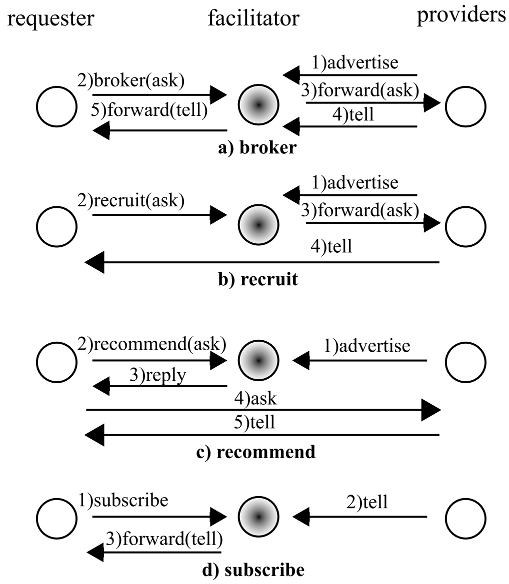


Fig. 4 Facilitation protocols.

```
Message ::=<msg_type>"("<cap_info>")"["("<trust_info>")"]
<msg_type> ::= REGISTRATION | QUESTION | ANSWER
<cap_info> ::=<context_type>"("<condition>*")
                    ("<extract_term>") ("<result>*")"
<trust_info> ::= [<option>]("("<agent_name><trust_value")")
<option> ::= DIRECT_ONLY | POSITIVE | NOT_NEG | ALL
```

Fig. 5 Message format.

type is parsed at the facilitator and the rest are parsed either at requesters or providers. Trust information includes option and pairs of agent name and trustworthiness. Option means degree of using trustworthiness on facilitation and it can be requested by both requesters and providers. The choices are:

- DP only
- DP and IP
- Not negative (that is, DP and IP and UN)
- All

Registration of trustworthiness is carried out by using the "advertise" performative of KQML. For provider agents, registration can be done with registration of capabilities as shown in **Fig. 6** (a), in which provider agent (agentA) advertises to the facilitator1 its capability "car_sales" and registers trustworthiness; that means it trusts agentB and distrusts agentC. For requester agents, registration of trustworthiness can be done with a request for facilitation as shown in Fig. 6 (b), in which agentB requests for brokering to the facilitator1 with conditions that the year be newer than 1997 and price be less than $9,000. This

```
a) advertise
  :sender agent A :recilitator 1
  :reply-with labell
  :content (REGISTRATION(car_sales()()())
           ((agentB P)(agentC N)))

b)broker-all
  :sender agent B :receiver facilitatorl :reply - with labell
  :content (QUESTION(car_sales((>year 1997)(<price
           9000))(MAkePrice)())(POSITIVE(agentCP)))

c)tell
  :sender facilitator1 :receiver agentA :in-reply - to labell
  :content (ANSWER(car_sales()())
           ((Toyota80000)(FORD7050)))
```

Fig. 6 Examples of KQML message.

also means the agent expects Make and Price as a result and, for trustworthiness, registers that it trusts agentC and wants to get only directly trusted agents.

Shown in Fig. 6 (c), an example of an answer from the facilitator1 means the results are a Toyota, which costs $8,000 and a FORD, which costs $7,050.

3.2 Trust Propagation Mechanism

Inside the facilitator, filtering potential partners is performed based on requested capabilities and trustworthiness. For maintaining registration in the facilitator, capabilities and trustworthiness are stored in a dynamic database inside the facilitator. In the database, trustworthiness values are stored by an n*n matrix, in which the n is the number of registered agents, for each capability type.

Trustworthiness registered by users could be either DP or DN. When a facilitator receives data, cells of the matrix which remain UN may be converted into IP or IN by calculation. Every time the facilitator receives new data, it recalculates all of IP, IN, UN data. Calculating indirect reputation (IP or IN) of one cell, from agent X_1, to agent X_n is done by the following two steps:

(1) Find paths from X_1 to X_n based on the following policies (**Fig. 7**):
  - An agent can use only the direct reputation of other agents
  - From X_j to X_{j+1} $(1 < j < n-2)$, only DP can be used.
  - From X_{n-1} to X_n, both DP and DN can be used.

(2) Tie-breaking if more than one path exists based on certain rules such as:
  - Majority decision: scoring for each path based on certain criteria such as its length and comparing the scores of positive and negative paths compare the total score of positive paths and negative paths.
  - Risk evasion decision: IN if at lease one negative path is found.

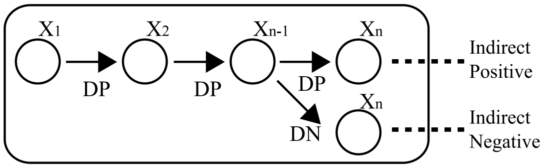


Fig. 7 Indirect reputation.

As far as the computational complexity for calculation is concerned, the worst case is to find all possible paths and compare them, for all cells of the table. In this case, complexity is:

$$(number\_of\_cells)*(number\_of\_possible\_paths) = (n*n)(1+(n-2)+(n-2)(n-3) + (n-2)(n-3)\dots 1) = O(n^n)$$

This method is not feasible as it is NP complete. Thus, to reduce the complexity, following two heuristic methods was considered:

(1) Method-1: Searching within limited length of paths, and

(2) Method-2: Finding the shortest path.

Method-1 is to find paths which have less than the specified length, and to do tie-breaking. The complexity of this method was:

$$(n*n)(1+(n-2)+(n-2)(n-3)+(n-2)\dots (n-q)) = O(n^{(q+1)})$$

Where q is the specified length.
Thus, feasible complexity can be achieved if q is small number.

Method-2 is to find the shortest paths and do tie-breaking by majority or risk evasion decision if more than one shortest path exist. The way of finding the shortest paths is as follows:

(1) Making k_th $(1 < k < n)$ reachable matrixes (RM) which have n*n cells, which show reachability from node i to node j within k steps. Let $V_k(i,j)$ be the value of the (i,j) cell of Kth RM.
  - Making 1st RM,
    $V_1(i,j)=1$ where $i=j$, or $(i,j)=$DP
    otherwise 0
  - Making $k+1$_th RM
    $V_{k+1}(i,j) = 1$:

where $\sum_{x=1}^{n}\{V_k(i,x)*V_1(x,j)\} > 0$

otherwise 0

(2) Calculating indirect reputation by using k_th RM.

- For each cell (i, j), finding all the nodes e[x] which have direct connection (DP or DN) to node j and calculate the length of shortest paths from node i to each e[x] by using RM.
- Do tie-breaking if more than one shortest paths exist (majority decision or risk evasion decision).

Thus, the complexity of Method-2 is:

$(Complexity\_of\_making\_nth\_RM)$
$+(number\_of\_cells)*(number\_of\_previous\_nodes)$
$= (number\_of\_cells)*(calculation\_for\_V_k(i,j))$
$+n^2*(n-2)$
$= O(n^4) + n^2*(n-2)$
$= O(n^4)$

Then each of two methods can be used based on the application domain where the facilitator is applied.

4. Implementation

We have implemented the facilitator as a JATLite-based multi-agent system, developed at Stanford University [9]. JATLite is a Java-based platform and consists of a message router for exchanging messages between agents and a template for developing agents which speak the KQML language. All messages are exchanged through the message router, as shown in **Fig. 8**.

Figure 9 shows a GUI of requester agents. In the figure, a request for facilitation with trustworthiness is described in the content parameter at the bottom left of the window and the answer forwarded by the facilitator originating from the provider is described at the bottom right of the window.

5. Facilitator Use

Take supply chain coordination at a construction project as an example. Recently, construction projects are carried out by general contractors who get an order and coordinate subcontractors who actually do the work. Consequently, a network for Project Supply-Chain coordination, where negotiation of task and schedule are performed, has been established, as shown in **Fig. 10**.

The project supply-chain coordination re-

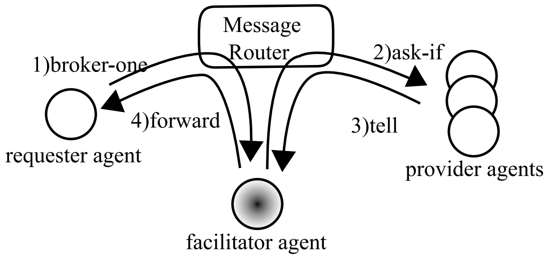


Fig. 8 JATLite and facilitator.

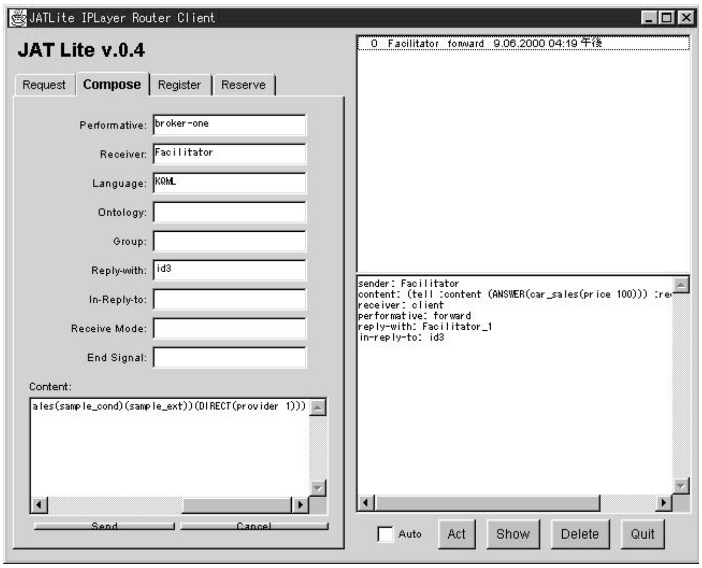


Fig. 9 Broker-one request example.

quires the collaboration of numerous suppliers and subcontractors. In particular, the degree of collaboration, including the sharing and joint creation of extensive information as well as the sharing of risks and benefits in the face of uncertainty, requires that the collaborators have a degree of mutual trust. The facilitator will help participants to form and maintain mutual trust information through trust- based facilitation for the project supply chain coordination.

The facilitator generally provides participants with opportunities to seek capable and trustworthy partners with whom they want to work. In cases of external changes in construction projects, which are ubiquitous in construction, the participants could seek outside partners to alleviate their losses. The facilitator could provide a longer list of eligible partners through the trust propagation mechanism than current practices where each participant maintains its list of eligible partners respectively.

As an example, suppose there is one general contractor (GC) and five subcontractors (SUBs). Suppose that the GC wants to subcontract some portion of its work — C3 — to a selected subcontractor, as shown in **Fig. 11**.

The GC wants to work with a capable and trustworthy subcontractor for the job. Therefore, the GC wants to ask the Facilitator to provide a list of eligible subcontractors by sending a message, such as QUESTION (C3&POSITIVE), to the Facilitator. The Facilitator checks the Capability Table and finds four capable subcontractors, such as SUB_B, SUB_C, SUB_D, and SUB_E. Among these SUBs, the Facilitator checks the Trust Table and finds three trustworthy SUBs — SUB_ B, SUB_C, and SUB_D — which have POSITIVE trustworthy values for the GC. Before sending a list of these subcontractors to the GC, the facilitator checks the Trust Table again and finds that SUB_B and SUB_D do not want to be introduced to the GC, based on their NEGATIVE trustworthy values to the GC. Therefore, the Facilitator reports only one eligible subcontractor by sending a message, such as ANSWER (SUB_C), to the GC. Then the GC negotiates with SUB_C for the job.

The Facilitator calculates and keeps trustworthiness values when the GC and SUBs register capability and provide DP or IP values for others with whom they have direct experiences. Default UN values will be changed to IP or IN through the Facilitator's trust propagation mechanism. For example, for the GC, the Facilitator tags IP for SUB_B because trustworthy SUB_A trusts SUB_B; IP for SUB_C because trustworthy SUB_D trusts SUB_C; and UN for SUB_E because trustworthy SUB_A and SUB_E have no information about SUB_E. Note that SUB_D does not trust GC even though GC trusts SUB_D. The trustworthy values are subjective for each one. Because of that, SUB_B has IN value to the GC. The unknown values of SUB_E will be changed after the GC evaluates SUB_C because both SUB_C and SUB_E trust

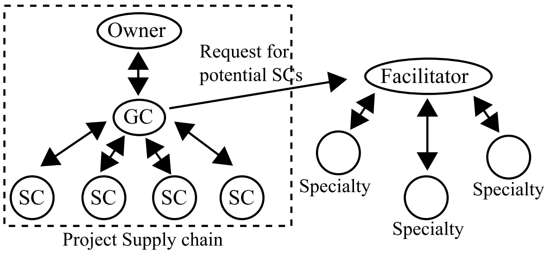


Fig. 10 Agents in project supply chain.

capability Table

	C1	C2	C3	C4	C5
SUB_A	V			V	
SUB_B		V	V	V	V
SUB_C	V		V		V
SUB_D	V	V	V	V	
SUB_E	V	V	V	V	V

Trust Table for C3

	GC	SA	SB	SC	SD	SE
GC		DP	IP	IP	DP	UN
SUB_A	DP		DP	UN	DP	UN
SUB_B	IN	DN		DN	DP	UN
SUB_C	IP	DP	IP		IP	DP
SUB_D	DN	IP	DP	DP		UN
SUB_E	UN	IP	DP	DP	UN	

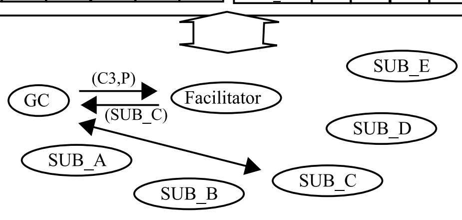


Fig. 11 Trust-based facilitation.

each other. Note that the GC has trusted only SUB_A and SUB_D before this facilitation process. With the aid of the facilitator, the GC will know more trustworthy SUBs than before. The more facilitating process will enrich the value of the facilitator, which means that the facilitator could suggest more eligible partners.

6. Conclusion

We propose a facilitator which finds capable and trustworthy partners on behalf of client users. We believe that the facilitator is the first trust-based facilitator which uses private-distributed-subjective-qualitative trust information.

There are some limitations we should mention. First, we should devise a more reliable algorithm for trust maintenance in terms of consistency, simplicity and relevancy. We also should add the self-healing mechanisms against malicious agents and support for newcomers to the community. And in order to be used in practical applications, we need more evaluation of this facilitator.

Acknowledgments Our thanks to Roxanne Zolin and Martin Ekstrom at Stanford University for giving us comments on our draft paper.

References

1) Better Business Bureau. http://www.bbb.org
2) Cafferky, M.: Free Word-of-mouth Marketing Tips home page. http://www.geocities.com/WallStreet/6246/quate6.html
3) Chavez, A. and Maes, P.: Kasbah: An agent Marketplace for Buying and Selling Goods, *Proc. PAAM'96*, London, UK (Apr. 1996).
4) CORBA Specifications. http://www.corba.org
5) Dun and Bradstreet. http://www.dnb.com
6) eBay. http://www.ebay.com
7) FIPA Specifications. http://www.fipa.org
8) Jini Specifications. http://www.sun.com/jini/
9) Jeon, H., Petrie, C. and Cutkosky, M.R.: JATLite: A Java Agent Infrastructure with Message Routing, *Internet Computing* (Mar.-Apr. 2000).
10) Manchala, D.W.: E-Commerce Trust Metrics and Models, *IEEE Internet Computing*, Vol.4, No.2, pp.36–44 (2000).
11) Labrou, Y. and Finin, T.A.: Proposal for a new KQML Specification, TR CS-97-03, Computer Science and Electrical Engineering Department, University of Maryland Baltimore County, Baltimore, MD (1997).
12) Sycara, K., Lu, J., Klusch, M. and Widoff, S.: Matchmaking among Heterogeneous Agents on the Internet, *Proc. AAAI Spring Symposium on Intelligent Agents in Cyberspace*, Stanford, USA (1999).
13) Zacharia, G. and Maes, P.: Collaborative Reputation Mechanisms in Electronic Marketplace, *Proc. HICSS-32*, Hawaii (1999).
14) Zolin, R., Fruchter, R. and Levitt R.: Building, Maintaining and Repairing Trust In Global AEC Teams, *Proc. ICCCBE-VIII*, Stanford, CA, ASCE, pp.874–888 (Aug. 1999).

(Received May 28, 2001)
(Accepted November 14, 2001)

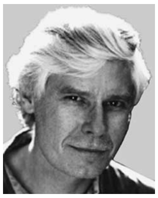
Chihiro Ono received the B.E. degree of Electrical Engineering, the M.S. degree of Computer Science from Keio University, Japan, in 1992 and 1994 respectively. Since joining KDD in 1994, he has been working on network management systems, database systems, and agent systems for electronic commerce. From 1999 to 2000, he was a visiting researcher at Stanford University. He is currently a research engineer of Network Service Lab. in KDDI R&D Laboratories Inc. He received Best Paper Award for Young Researchers of the National Convention of IPSJ in 1996.

Dai Kanetomo received the B.E. and M.E. degree of Administration Engineering from Keio University, Japan in 1994 and 1996 respectively. Since joining NTT, he was working on object oriented systems and software agent technologies. From 1999 to 2000, he was a visiting researcher at Stanford University. He is currently working on developing new business and service based on alliances with foreign companies at NTT Comware Co.

Keesoo Kim is a Ph.D. Candidate in Construction Engineering and Management Program, Department of Civil and Environmental Engineering, Stanford University. His Ph.D. research has been focusing on distributed coordination of project participants in construction projects using agent technologies. His research interest includes application of trust information to construction supply-chain management. Prior to this, he has worked for Daewoo E&C Co., Ltd for nine years in domestic and international construction projects. He earned B.S. (1985) in Architectural Engineering from Seoul National University, Seoul, Korea, M.S. (1996) and Engineer (1997) in Civil and Environmental Engineering from Stanford University.

Boyd C. Paulson, Jr. is the Charles H. Leavell Professor of Engineering in Stanford's Graduate Program in Construction Engineering and Management. He was a Visiting Professor at the University of Tokyo, the Technical University of Munich, the University of Strathclyde in Glasgow, Scotland, and the University of Hawaii. He earned his B.S., M.S., and Ph.D. degrees in Civil Engineering from Stanford University. He is the author of two books and over 100 papers. His research and teaching interests focus on computer applications in construction, construction equipment and methods, international construction, and affordable housing. He has had numerous sponsored research projects in these and other areas and has received several national and international awards for his work. He has studied construction projects and organizations in visits to numerous countries in six continents, including fellowships and grants to support research and teaching in China, Germany, Japan, and the UK.

Mark Cutkosky is a principal investigator of the Design Interface for 3D manufacturing and Biometric Robotics at the Center for Design Research (CDR), director of the Dextrous Manipulation Lab, and co-director of the Stanford Alliance for Innovative Manufacturing at Stanford University. He joined the Design Division of the Stanford Mechanical Engineering Dept. in 1985 after working for several years in the Robotics Institute at Carnegie-Mellon University and as a machine design engineer at ALCOA in Pittsburgh, Pennsylvania. He received B.S. of Mechanical Engineering at University of Rochester (1978), M.S. (1982) and Ph.D. (1985) of Mechanical Engineering at Carnegie-Mellon University.



Charles J. Petrie, Jr. is the founding Executive Director of the Stanford Networking Research Center (SNRC), and a Sr. Research Scientist, Stanford Computer Science Department (pending). Prior to this, he was a Senior Research Scientist at the Center for Design Research (CDR) at Stanford University. He was the founding Editor-in-Chief of the IEEE Internet Computing Journal. Prior to Stanford, he was a founding member of the MCC AI Lab where he led the Proteus project, which had the first commercial use of AI software from that lab, and founded the ongoing International Conference on Enterprise Modeling Technology (ICEMT). He has organized many workshops and conferences and is often an invited speaker. He has published extensively on inferencing architectures and the use of truth maintenance for concurrent engineering design, developing the seminal Redux model of design. He is co-developer of the widely used JATLite agent platform. His current research interest is the dynamic revision of web services using agent technologies. He has a Ph.D. degree in Computer Science from the University of Texas at Austin. He is a member of AAAI, IEEE Computer Society, and ACM.