Smart Distance for Information Systems: The Concept

Yiming Ye, Prabir Nandi and Santhosh Kumaran[1]

Abstract—**We propose the concept of a "smart distance" for information systems and illustrate how to use it to interwoven various dynamic and heterogeneous elements of the system. With "smart distance" infrastructure operating within an information system, it can eliminate inefficiencies that are due to lags and latencies that exist in the traditional environment. On a larger scale, with "smart distance" operating across WWW, the Web can be more adaptive, useful, and popular.**

I. Background

THE pervasive connectivity of the Internet, coupled with an increasing distribution of organizations, are introducing profound changes in the way enterprises are set up and operated and are intensifying the forces within and across enterprises. To remain competitive under this environment, organizations need to move fast and to quickly adapt to business-induced changes. It must be able to sense the salient information, transform it into meaningful, quality business metrics, respond by driving the execution of business decisions into operational systems; and finally track the results against actions and expectations.

In parallel with this trend, there have been interesting developments in the fields of Intelligent Agents and Distributed Artificial Intelligence (DAI), notably in the concepts, theories and deployment of intelligent agents as a means of distributing computer-based problem solving expertise. Intelligent agents are well suited to the emerging character of the adaptive enterprise in which the distributed operations must be orchestrated into a synchronous flow.

There have been arguments or efforts to use agents or business objects to construct adaptive systems or enterprises [1] [4] [5] [8] [11] [12] [14]. For example, Hayes-Roth [8] studies the issue of how to build agents that function effectively in "adaptive intelligent systems" (AISs) that vary dynamically along dimensions like task requirements, different resources, contextual conditions, and performance criteria. She argues that an agent must adapt several key aspects of its behavior to dynamic situation such as its perceptual strategy, its control mode, its choices of reasoning tasks to perform, its choices of reasoning methods for performing those tasks, and its meta-control strategy for global coordination of all of its behavior. Despite various efforts in studying object oriented or agent oriented adaptive enterprises, as far as we are aware of, there are no working systems being widely used in practice. This is because that the state of the art of artificial intelligence has not reached to a stage such that an adaptive system can be operated effectively without human beings involvement.

In this paper, we propose the concept of a "smart distance" for complex enterprise systems and illustrate how this can be implemented by using Adaptive Document (ADoc) [12]. The idea is to wrap any business artifacts where "smart distance" is needed with a layer that enables this functionality. This layer is realized by extending an ADoc with relevant "smart distance" components. This will guarantee that the configurations and interactions of the elements of an enterprise system can be best placed, at any time and under any contextual environment. The concept of a "smart distance" comes from the observation that people like to adjust the "distances" among them when there are choices (for details, see [16]). The concept is further extended by including various elements in an enterprise system such as databases, business objects and intelligent agents, in addition to human beings. We define a "smart distance" in an organization as distances that are autonomously and adaptively adjusted based on contextual information with the goal that tasks can be best performed. Our approach can guarantee that the "distances" of various elements of an information system are adaptively placed to favor the task, at any time and under any contextual environment.

II. Smart Distance for Business Artifacts

Here we illustrate the concept of a "smart distance" for an enterprise system. As we have discussed before, this concept comes from the observation of human-human interaction. It is extended to agent supported collaborative work where intelligent agents are used to adjust the communication channels among people based on contextual information [16]. Here, we further extend the concept to an adaptive enterprise system to refer to the situation that different artifacts dynamically adjust their "distance" configurations such that the performance of the enterprise can be maximized. These artifacts can be users of the enterprise, business entity agents, and business objects, etc.

[1] IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA {yiming,prabir,sbk}@us.ibm.com

The "distance" refers to the various degrees of awareness, communications, and interactions among different artifacts.

The smart distance concept can be intuitively represented in Fig. 1. Smart distance between people refers to the situation where people intelligently adjust their distance based on various social contexts and preferences (Fig. 1(a)). Fig. 1(b) shows the concept of a "smart distance" between people and object. The person in the figure adjusts the distance between his eyes and the books under different contexts and vision conditions. Fig. 1(c) shows the natural distance, or the best distance between two artifacts under a contextual situation at a given time. Fig. 1(d) shows the real distance between two artifacts, which may not be the natural distance. Fig. 1(e) shows the situation that the two artifacts perform some autonomous actions such that their distance can be the same as their natural distance.

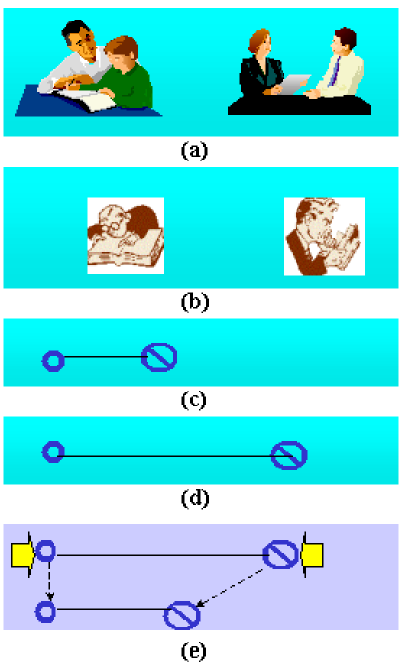


Fig. 1. Smart distance illustration.

The following is a formal definition for smart distance for an enterprise system.

- E is an Enterprise system
- $A = \{a_1, \cdots, a_n\}$ is the set of all the artifacts of this enterprise system. An artifact can be an object, an autonomous element, an intelligent agent, a database system, an information system, or a human being etc.
- For any two artifacts a_i and a_j, suppose that there are I_{ij} different kinds of interactions for a_i to interact with a_j. We call each kind of interaction a "channel". For example, people to people interaction through web can have video channel, audio channel, text channel.
- For a given channel c, there might be different degree of interactions; we used $|c| \in [0,1]$ to represent the degree of interaction. The bigger the value of $|c|$, the more intense the interaction for this channel. With $|c| = 1$ to represent the strongest interaction for this channel, and $|c| = 0$ to represent the weakest interaction through this channel. In most case, $|c| = 0$ means that this channel is closed. For example, the degree of interaction for a video channel from one person to another can be defined as the resolution and the update rates of the video transmission. The degree of interaction from a monitoring agent to a user can be defined as the frequency that the monitored data is sent to the user.
- The distance from a_i to a_j can thus be represented as a vector: $d_{ij} = \left\langle |c_1|, \cdots, |c_{I_{ij}}| \right\rangle$.
- The distance configurations (at time τ) for a given enterprise can thus be represented by a matrix:
$$D(\tau) = \begin{pmatrix} d_{11}(\tau), \cdots, d_{1n}(\tau) \\ \cdots \\ d_{n1}(\tau), \cdots, d_{nn}(\tau) \end{pmatrix}$$
- Under a given contextual/environmental condition at time τ , $\Omega(\tau)$ (it might contain many parameters), there exists a natural distance configuration (or called best placed distance configuration) $D_{natural}(\Omega(\tau))$. If the distance configurations of all the artifacts in the enterprise E is equal to this natural distance, that is, if $D(\tau) = D_{natural}(\Omega(\tau))$, then the performance is maximized.
- In most situations, $D(\tau) \neq D_{natural}(\Omega(\tau))$
- Smart distance for an enterprise system means that the artifacts in an enterprise act autonomously such that $\| D(\tau) - D_{natural}(\Omega(\tau)) \|$ is minimized. This is a challenging task.

Our goal is to construct the enterprise system such that all the artifacts will be woven by "smart distance". Since usually best distance configurations are very difficult to be pre-determined/pre-coded because of the complexity and the dynamics of the environment, we need to provide a way of adjusting these natural distances easily.

III. Example Scenario

We use the following example to explain how smart distances among different enterprise users can be implemented with Adaptive Document (ADoc) [12]. An ADoc is a business object that is implemented using Enterprise Java Bean. It has state machines inside the bean to specify its states under various contextual situations. It also provides a convenient GUI for user interaction.

Suppose that a big retailer store has a virtual information system to process the supplier related issues. Suppose that there are a large number of distributed human agents associated with the retailer to handle the contract issues with various suppliers. Whenever a supplier has a product to offer, it will log onto the system and input supply related information. Then a human agent from the store will take over the issue and negotiate with the supplier in order to have the contract signed. Since there are a large number of human agents for the store, there is no way for the human agents to track each other's activity and to interact among each other. However, these may not be good to the retailer because appropriate interactions among human agents might help the retailer to do better deals under certain contextual situations.

For each user agent, we construct an ADoc to represent it. Within the ADoc, there is a business process template pool that provides all the different business processes that this user agent is associated with. For example, the left side of Fig. 2 shows a simple business process that a contract manager might be involved. First, a supplier logs onto the information system requesting to supply a product. Then, the contractor and the supplier will discuss about the contract and make a series of modifications. When the draft is finalized, it will be sent to the upper management chain for approval. The results will be that either the draft is approved or rejected.

Awareness among contract managers is needed during the

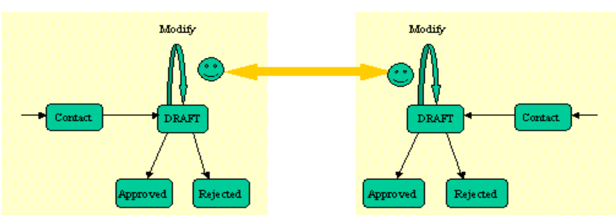


Fig. 2. Smart distance between contractors of the retailer store. The state machines represent the business flows of the contracting process.

"Draft" stage. Because it is possible that the two contracts in progress might be related to the same product. Thus, from the enterprise point of view, the two managers should be aware of each other's activities and interact with each other so as to get the best deal in the contracts (Fig. 2). This functionality is realized by extended ADOC (corresponding to the "Smart Distance" module of the BESA agent architecture) and by the "Smart Distance" directory of the enterprise server.

IV. Local Registry

Whenever a human agent logged onto the system, an ADoc is generated to represent the user. This user will then input through the ADoc UI to specify his awareness requirements, or in other words, his "smart distance" requirements under different contextual situations. This ADoc will track the user's activities and provide the "smart distance" functionality based on the user's registration. The local awareness registry will contain a Hash table. Each item of the table gives one awareness requirement that contains a list of information. Fig. 3 shows a specific example UML representation of the local registry for smart distance for the current example. It is an extension of ADOC [12]. The distance values in this example are: no awareness, Sametime [9] interaction, Phone interaction, Check document, and Meeting scheduling. Please note that Fig. 3 only gives a specific model of a local registration. In general, different systems may model the local registrations quite differently. The following list explains the details of the most important elements for a generic smart distance extension with respect to a single registration item.

- Awareness Condition for the current agent. This entry gives the condition to start the Smart Distance process, or in other words, the condition to start choosing distance values.
  - BPT_ID: Business Process Template ID. For example, the Template for Contract business process as illustrated in Fig. 2. In Fig. 3, it is represented by "My State Machine ID" of the "AW Condition Class".
  - State_ID: the state for the BPT_ID. For example, the "DRAFT" state in Fig. 2. In Fig. 3, this information is represented by "My StateID" of the "AW Condition Class".
- Condition on "when" to interact with "which" agents:
  - AE_Category: the category of the other agent to look for. In the above example, the category of the agent to look for is also a contract agent. In Fig. 3, this is represented by "HisAgentID" of the "AW Condition" class.
  - BPT_ID: the business process that the other agent is currently in. In Fig. 3, it is represented by "HisStateMachineID".
  - State_ID: the state of the BPT_ID the other agent is at. In Fig. 3, it is represented by "HisStateID".
- Actions: this gives the awareness actions list to be taken. These actions are distance values. For each action, the following information needs to be registered. There might be more than one action to be taken.
  - Awareness_ID. This gives the category of awareness to be requested. The list of awareness categories is listed in the global server that will be discussed in the next section. One example of the awareness is to provide a communication channel such as open a SameTime [9] window for interaction. Some awareness categories might be quite complex and may even contain the interaction choreography or conversational policies and protocols that specify a structured interaction. In Fig. 3, the awareness selections is given by the "AW ActionList" which provides all the awareness

requirements that need to be satisfied. Please note that when the Cardinality is “0”, then the corresponding awareness channel will not be opened.

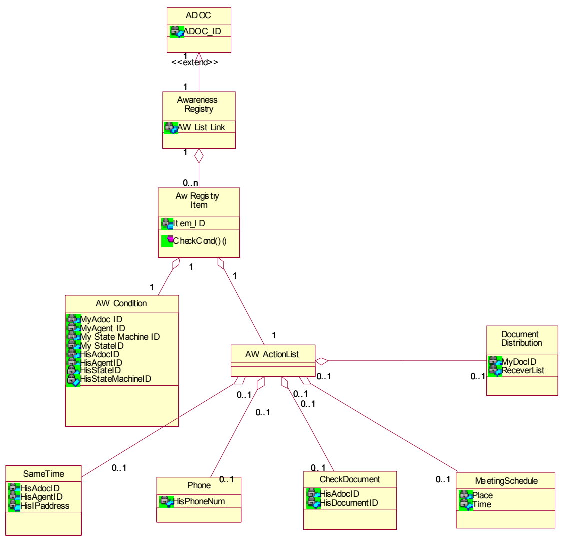
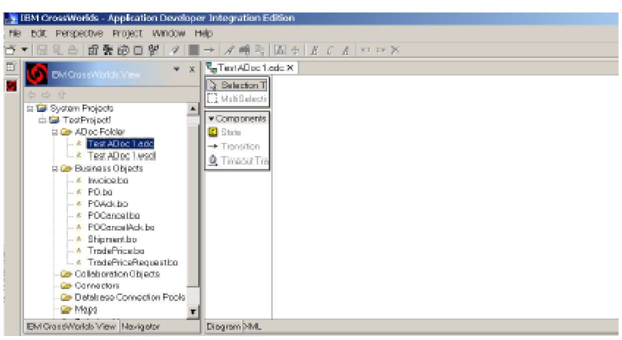
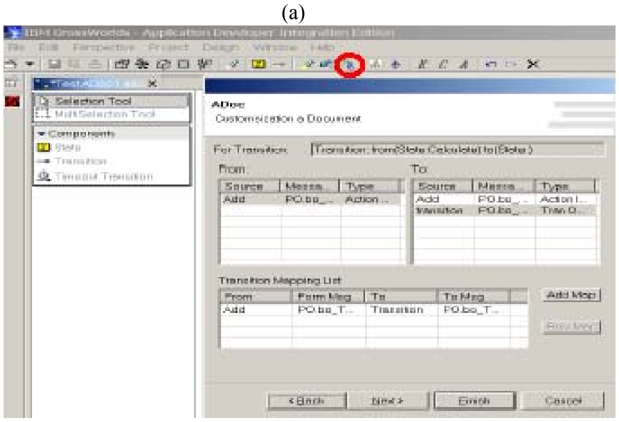


Fig. 3. UML representation of the local registry.

To implement the extension of ADoc to handle smart distance, we can either directly modify the code or use the ADoc builder that is currently being developed [17]. Fig. 4 shows the “state machine editor” and the customization Wizard of the ADoc builder.



(a)



(b)

Fig. 4. (a) ADoc state machine editor. (b) ADOC customization wizard.

V. Global Awareness Server

The awareness functionality during the runtime is realized by a Global “Smart Distance” Awareness Server. The local registration of an agent gives its user’s preference for distances. However, sometimes the enterprise might want to control the awareness level among various agents from a global point of view. For example, one user might want to interact with his director at a certain state. However, the enterprise might not allow this interaction because the user should interact with his/her first line manager for the corresponding issue. Thus, based on the contextual condition, the Global Smart Distance Server can enable individual awareness requests and also disable individual awareness requests.

Whenever a user agent logs onto the system, an ADoc will be generated to represent the user. After the user input the Local registration information, it will be transferred to the Global Registry. The Global Awareness Server collects all the Local Registration information from all the user agents. During the run time, the Global Awareness Server tracks the activities and performs matchmaking constantly so as to enable awareness requests when conditions are met.

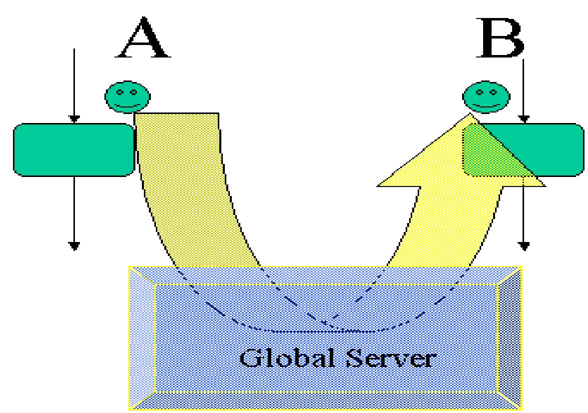


Fig. 5. Global server controls the degree on the channel from agent A to agent B during the run time.

Here is a simple example illustrating how the smart distance mechanism works during the run time. We assume the same store and the same virtual information system as described in Section 3. We also assume that the Global Awareness Server has a rule specifying that a communication channel will not be open if it only satisfies one agent’s requests.

1. A supplier logs into the system, wants to supply tables to the store
2. John, a contract agent for the store, takes the job and signs in.
3. A contract ADoc is generated with ID_1, for the purpose of representing John.
   a. ADoc ID = 1
   b. John, as an agent, get an agent ID = A1
4. Through the ADoc User Interface, John inputs the following information into the Local Registration.

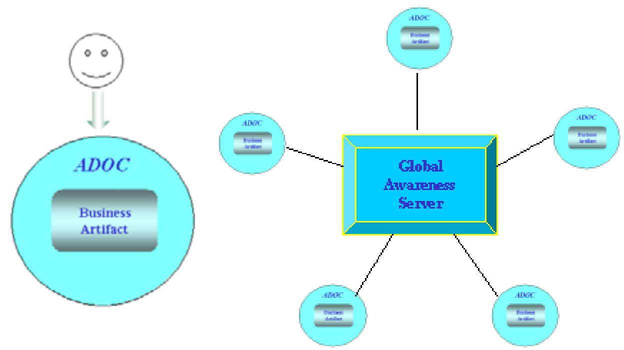
- a. Condition
    - i. John's Info:
        1. ID=A1
        2. State="Draft
        3. Object="Table" (the product from the supplier).
    - ii. Other Agent's info (the information on other agents that adaptive communication might be needed):
        1. ADoc ID = x (x means that any ADoc will be considered).
        2. Agent ID = x (x means any agent will be considered)
        3. State="Draft"
        4. Object="Table"
- b. Action
    - i. Open sametime chat window for both agents if they are not in the same building
    - ii. A phone call to schedule a face to face meeting if the agents are in the same building.

5. John's registration is propagated to the Global Registry. Global registry server will track John (A1)'s status and will check the condition info. whenever John's status is updated.
6. A supplier for "chair" logs into the system.
7. Mary takes the job and signs in.
8. An ADoc document ID_5 is generated and an Agent ID=A5 is generated for Mary.
9. Mary registered the local registry. She requests a sametime communication if there is another agent that is in the "Draft" state during a contract process and if the contract is related to either tables or chairs.
10. John and Mary are both at the Draft state. This is detected by the Global Awareness Server.
11. The communication channel for John and Mary does not open because only Mary's condition is satisfied (John wants interactions only when the other party is also doing a contract related to tables).
12. Another supplier for "table" logs onto the system.
13. Peter, who is at another site of the store, takes the job.
14. An ADOC is generated with ID_7 for Peter.
15. Peter registers the exactly same awareness condition as John.
16. Peter proceeds to the Draft state.
17. The Global Awareness Server responds to John's and Peter's requests and opens the sametime window for their communication.

VI. Discussions

In this paper, we propose the concept of a smart distance for complex enterprise systems and illustrate how to use ADoc to interwoven various business artifacts and to realize this functionality.

Our idea is to use ADoc to wrap any business artifacts that need "smart distance" functionality (Fig. 6(a)). These artifacts can be any elements of the system, ranging from business objects and databases to enterprise users. The UI functionality of the ADoc makes the adaptive change of distance requirements under dynamic, changing contextual situations possible. For ease of illustration, the scenario discussed in this paper is related to provide "smart distance" among users. However, the approach can be used to enable "smart distance" among various business objects within and across information systems. An information system with "smart distance" connecting its business artifacts can be more adaptive and on demand. On a larger scale, if we consider the World Wide Web as an ubiquitous information system, then providing smart distance among the huge number of artifacts of the WWW will make the Web even more useful, adaptive, and popular. Smart distance over the WWW can provide not only adaptive social networking among Web users, but also adaptive "social networking" among the huge number of Web artifacts. One approach to enable this is to build standards on "smart distance" over Web artifacts such that Web users can have a common language to specify their "smart distance" requirements that are understandable to programs or agents. We predict that this will be a reality in the future.



(a) (b)

Fig. 6. (a) Use ADoc to wrap a business object in an enterprise. The UI of ADoc makes the change of distance requirement possible. (b) "Smart distance" is enabled with a Global Awareness Server. Please note that ADoc is only one way of realizing "smart distance". Many other approaches can be used. For example, we may use Enterprise Java Beans to implement the functionality.

Acknowledgement

The authors would like to thank David Cohn for his support on the idea proposed in this paper and for his valuable suggestions on the possible application of this idea to business process monitoring. The authors would also like to thank Michel Desmarais for his valuable comments on the content and writing and Jiming Liu for his suggestions on the re-organization of this article.

REFERENCES

[1] J. Bigus, The Agent Building and Learning Environment, Proceedings of International Conference on Autonomous Agents, June 3-7, Barcelona, Spain, Pages 108-109, 2000.

[2] M.L. Brodie. The emperor's clothes are object oriented and distributed. In M.P. Papazoglou and G. Schlageter, editors, Cooperative Information Sys-tems: Trends and Directions, pages 15–48. Academic Press, 1998.

[3] Simon Cheng, Mathews Thomas, Santhosh Kumaran, Amaresh Rajasekharan, Fred Wu, Yiming Ye, Ying Huang. A Model-driven Approach for Item Synchronization and Uccnet Integration in Large E-Commerce Enterprise Systems, 5th International Conference on Enterprise Information Systems, Angers, France, 23-26, April 2003.

[4] P. Eeles and O. Sims. Building Business Objects. John Wiley & Sons, New York, 1998.

[5] B. Grosz, L. Hunsberger, and S. Kraus, Planning and Acting Together. AI Magazine, 20(4):23-34. 1999.

[6] James E. Hanson, Prabir Nandi, and David W. Levine, Conversation-enabled Web Services for Agents and e-business, Proceedings of the International Conference on Internet Computing (IC-02), CSREA Press, 2002, pp.791-796.

[7] J. Hendler, and A. Agrawala. Mission critical planning: AI on the MARUTI real-time operating system. In Proceedings of the Workshop on Innovative Approaches to Planning, Scheduling, and Control, 77-84, November, 1990.

[8] B. Hayes-Roth, An architecture for Adaptive Intelligent Systems, Artificial Intelligence Journal, 1993?

[9] http://www.lotus.com/products/lotussametime.nsf/wdocs/homepage

[10] http://www-3.ibm.com/software/integration/holosofx/1126announce.html

[11] Maamar, Z. and J. Sutherland. Toward Intelligent Business Objects: Focusing on techniques to enhance BOs that exhibit goal-oriented behaviors. Communi-cations of the ACM 43(10): 99-101.

[12] Prabir Nandi, Santhosh Kumaran, Terry Heath, Raja Das, Kumar Bhaskaran, ADoc-Oriented Programming, The 2003 International Symposium on Applications and the Internet (SAINT'2003), Jan 27-31, 2003, Orlando, Florida, USA.

[13] Papazoglou, M. Agent Oriented Technology in Support of E-Business: Enabling the Development of "Intelligent" Business Agents for Adaptive, ReusableSoftware Communications of the ACM, 44(4):71-77

[14] J. Sutherland, W.J. Heuvel (2002), Enterprise Application Integration Encounters Complex Adaptive Systems: A Business Object Perspective. Proceedings of the 35th Hawaii International Conference on System Sciences , Hawaii, USA.

[15] Wu, L.S.Y., Hosking, J.R.M., and Doll, J.M., "Business Planning Under Uncertainty: Will We Attain Our Goal?", IBM Research Report RC 16120, Sep. 24, 1990

[16] Y. Ye, S. Boies, P. Huang, J. Tsotsos. (2001). Agents-Supported Adaptive Group Awareness: Samrt Distance and WWWaware. IEEE Transactions on Systems, Man, and Cybernetics. Part A: Systems and Humans, Vol. 31, No. 5. pp: 369-379.

[17] Jun Zhu and Zongwei Luo, ADoc Builder Manual Draft, 2002.