H-0249 (H0503-003) March 9, 2005
Computer Science

IBM Research Report

Towards a Real-Time Ad-Hoc Intelligent Collaboration in a Pervasive Environment

Uri Shani, Ziva Sommer, David Botzer
IBM Research Division
Haifa Research Laboratory
Mt. Carmel 31905
Haifa, Israel

Towards a Real-Time Ad-Hoc Intelligent Collaboration in a Pervasive Environment

Uri Shani, Ziva Sommer, David Botzer
IBM Haifa Research Lab
shani@il.ibm.com

Abstract

We propose an intelligent collaboration service we call ICS which integrates several technologies and methodologies to enable ad-hoc collaboration among parties in a certain organization or working place. We use a rule-based situation management system capable of real-time massive event processing which accepts events to drive a notification mechanism, in order to establish collaborative situations among parties, within a pervasive environment. Looking at the medical care environment, we present a case study, provide a rationale and offer a problem model for this case in which an ICS can be deployed. Based on that, we assess the technological feasibility and the need, which can be also applied onto other domains which would be even more demanding than the medical health environment.

1. Introduction

Event notification (EN) is in effect a generalization of the publish-subscribe programming pattern, application model and protocol [1] . EN consists of event detection and a notification mechanism which identifies targets for notifications, and generates contents for them. EN also has a mechanisms to transport these notifications to their targets. EN has been studied extensively in the past decade, and common application environments would most interestingly include wide-area networks [2] , the Internet, which presents a very large and open domain [3] , and mobile networks of ad-hoc devices, which present additional challenges [4] .

In this paper, we are looking as a base at INS (Intelligent Notification System) [5] . As depicted in *Figure 1*, the INS EN consists of participants having three different roles: Provider of events, a Criteria application that detects events against subscriptions, and a Delivery of notifications to targets according to the subscriptions. Both targets of notifications and providers of events are usually tier-0 devices (workstations or hand-held mobile devices). Examples of INS providers are Internet stock-market tickers, events in a calendar application, or an instant messenger, enterprise databases (tier-2), etc. Targets may too include mobile devices.

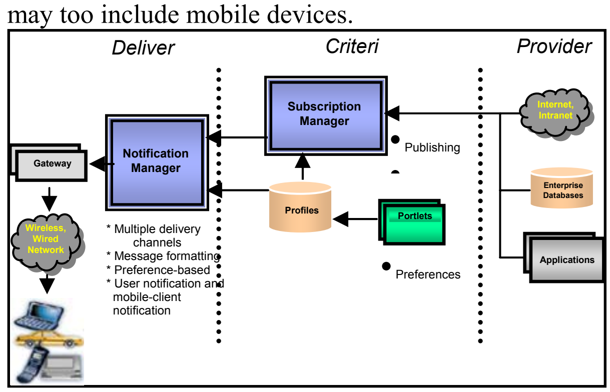

Figure 1: INS 3-part major components

The events and notifications are transmitted over networks and their support platforms such as the Internet, the wireless mobile networks and switches, mail services, instant messages and so forth. The event notification core logic is in the Criteria part which is a server application (tier-1) in which events are detected, and from which notifications are delivered.

Events as messages play a central role in application middleware technology and has supporting standards such as the Java Messaging Services API (JMS) [6] , or the OMG Notification Service Specifications [7] . Infrastructure for events delivery in both the local and global scales are for instance IBM MQ-Series [8] , and the Microsoft Herald Project[9] . The logic in the notification engine of EN systems is one of the most important components in EN systems (see for instance [2] and [2] [3]). Mobile pervasive environments offers also location information for events, or rather the devices which can create events and receive notifications [10] .

Notifications may be a simple SMS message sent to a mobile device, or can in-fact be a complex service such which can get parties to collaborate. Collaboration among parties presents a new challenge to NE systems, and a mechanism to do that, called UCOMMIT [11] has been developed for the UNITE EU FP6 project.

In the taxonomy for CSCW (Computer Supported Collaborative Work), Shani [15] presents a thorough look into the different types of collaborations which may arise in various working environments, supported by a variety of IT solutions. Ad-hoc collaboration would be one which arises from dynamic circumstances which cannot be foreseen. An effective collaboration can be obtained and be managed based on real-time events which reflect real situations. These are complex events (see for instance a report on that in a related network management domain **Error! Reference source not found.**) that reflect situations which have a life-span and in which many events may be involved. A mechanism to process such complex events and manage long-lived temporal situations is the AMiT (Active Middleware Technology) system [13] .

In this paper we focus on domain where environments consist of hand-held, mobile wireless devices using to communicate in various ways among parties and members of a certain organization. The proliferation of such environment is very large and growing constantly. Organizations may be geographically spread on wide areas, or share a common building where distances are short. When within a certain building, parties tend to collaborate in person in a more effective way than via devices. To support such ad-hoc collaborations, we offer a system that extends the INS architecture with several additions which include location information, complex event processing, and collaboration-support mechanism to make it an ICS (see **Figure 4**). We evaluate the need and complexity of the solution in a medical environment such as in a hospital.

2. The AMiT engine

AMiT [13] (Active Middleware Technology) is a lightweight and agile business rule engine with Complex Event Processing (CEP) capabilities. AMiT is based on the observation that in many scenarios, single events are insufficient, and rules using a complex composition of events, happening at different times and having different context, are required. For example, a client wishes to be notified on real time, when at least two (out of the four) stocks: IBM, Oracle, Sun, Microsoft, are up 5 percent since the beginning of the trading day.

AMiT receives two types of inputs (see **Figure 2**):

- The meta-data, which contains the definitions of the events and the rules to be detected.
- The event instances that are submitted from the external sources.

The events from different sources are fed to the rules

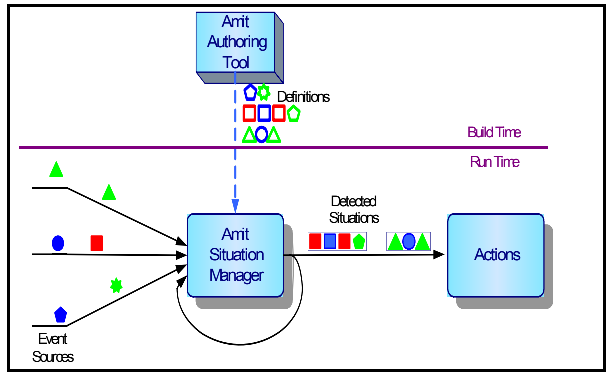

Figure 2: AMiT situation management of events

engine. There they are composed according to their types, temporal properties and semantic properties, as defined by the rules. A rule definition contains a life-span, the time interval during which situation detection is relevant, a composition operator, describing how the incoming events should be processed, and filtering conditions on the result of the processing. A detailed description of the AMiT language can be found in 0[14] .

The result of this processing is a situation – a new event in the system. This event contains information from the raw events which were used to generate it. It can be fed back to the engine again and participate in calculating new situations – "nested" situations – or sent out as an alert to the external application using the engine.

Within an EN system, the detected situations will activate notifications. As part of the metadata, the user can instrument a link with the delivery mechanism such as in INS as will be seen later on.

3. The ICS architecture

In basic terms, ICS does not differ much in architecture from the intelligent notification services INS. We see three additional components we rank from ‘essential’ to ‘recommended’. First and essential is the addition is AMiT. This will facilitate a memory about events in the form of situations with life-spans. That will allow for instance to extend the “one-way” notifications, which are also unreliable and may get lost (unless you trust the reliability of delivery mechanisms such as SMS for instance). In collaboration among human parties, events are delivered at the end to a human user, who may fail to detect an event and will not act upon it even if the entire delivery mechanism to his/her tier-0 device will work perfectly reliably. In the sequel we investigate the medical domain and show scenarios for our subject matter in this domain that makes this a "closed control loop" where consequences of notifications are fed back into the system in the form of new events which can be matched against the relevant situations to facilitate further resolute actions.

The second addition is a collaboration-support technology such as discussed in UCOMMIT [11] that is not essential for all our scenarios, but provides the missing link in many useful scenarios at the medical care domain we discuss next.

The third addition is location information. That will allow yet more powerful scenarios and thus is greatly recommended. In essence, this is not different than any other event information fed into EN system such as INS. For ad-hoc collaboration in the physical real world, one may imagine how this can serve to facilitate more intelligent notifications than without it.

A fourth component is recommended as it provides domain-specific information. In the medical domain, information about the patients' condition should be added in the Detection part to supply events about the subjects of care in healthcare institutes. In other domains that of course would be as appropriate. We include in our medical-oriented discussion here a detection element in which medical information of patients is collected and stored for data mining and analysis such as our ICS will require [16] .

4. Complexity of ICS Scenarios

For instance, a collaboration such as meeting in a conference room may be initiated by the system based on events coming from calendar sniffers, or due to rules based on a situation in the environment (i.e., new financial data, new customer deal has become "hot", a situation in the battlefield, or a critical lab-test of a hospitalized patient). The meeting includes a number of people, of which a subset is essential. These may be specific individuals, or people in certain roles and skills (i.e., customer account managers, financial analyst of a certain market segment, technical experts, or skilled medical personnel). The collaboration execution criteria are the actual gathering of a satisfactory group of people in a certain meeting room. That may be confirmed by a feedback on a certain terminal, or via contextual (presence) information fed into the system as events from the event generators.

This complex episode may use several complementing or even competing methods for a common goal. Once the expected common goal is satisfied, the system needs to ensure that all these competing means are notified and cancelled properly.

An ICS deployment presents a certain complexity which is measured along several dimensions.

- The number of clients or end-users connected with the system, for which profiles are defined
- The number and variety of end-devices which are targets of messages/notifications
- The variety and number of event providers. These could be pervasive devices or software tools such as calendars, databases, web crawlers, agents, and so forth
- The "density" of events generated by the environment, meaning number of events per unit of time.
- The types of scenarios may present simple notification cases up to complex episodes as described briefly above.
- The number of notification messages and on going situations density – meaning number of coexisting on-going situations in average and at peek times.
- Is there a critical need for real-time operation?

5. Target Deployments of ICS

There are certainly endless deployment targets for ICS, the following table presents a growing scale of complexity for several deployment domains.

Deployment	# Clients	# End Devices	# Event Providers	Event Density	Scenario Types	# Notifications and/or Situations	Real Time
Server farms	10-s	10-s	# servers	low	simple	100-s	Small
Hospital	100-s	100-s	1,000-s	high	complex	1,000-s	Some
Security	100-s	100-s	100-s	high	medium	100-s	All
Defense	1,000-s	1,000-s	1,000-s	Very-high	complex	1,000-s	High
Army	10,000-s	10,000-s	10,000-s	Extremely high	Very complex	10,000-s	High

6 The medical ICS environment

The medical environment presents many challenges which can be met by a proper ICS solution. We hereby show several relevant scenarios.

61. Notification Feedback

To enhance the base INS facilities of notification delivery, AMiT situation management is used to facilitate the processing of complete "notification situations," which ensure that the information notified has been duly processed and noted by the proper recipient. The first scenario uses an SMS delivery notification tool. The second adds qualitative information to the process, whereby personnel is identified by skills, (or roles), and by the situation's level of urgency.

- A physician requests an urgent lab test for a patient in critical condition.
- When the lab results are ready, SMS messages of lab-test results are delivered to the physician, repeatedly at a certain rate.
- The physician must confirm that she received the message by replying to this SMS (via portlets).
- If no reply is received within a pre-specified timeout limit, the message is sent to a different physician with the same skills (specializing in the same field), who can analyze the results.
- Delivery of the messages can be done according to skills, level of urgency, etc.
- Patient may also be in the loop and receive SMS notification.

Figure 3: Surgery room situation

6.2. Spontaneous (virtual) meetings and consultations

A physician examined a patient in a critical condition. The physician wants to consult with some colleagues, in order to get additional input. Say, for example, she must make a decision that requires the participation of at least three additional physicians, according to hospital regulations. Via presence awareness, physician location is notified to the system. Once a "quorum" is possible, a meeting to consult the matter can be held. All physicians receive notification through beeper/SMS that a meeting is being held.

6.2.1. Virtual meeting. Hospitals are media-rich environments, allowing physician to join such meetings through proper accessible terminals to form a virtual meeting – same time, different places.

6.3. Ad-hoc casting by skills for emergency

- In an emergency situation, the patient's conditions are fed into the system.
- Certain operation procedures require a specific set of surgeon skills.
- Through presence awareness, skills, and schedule information, the system can summon the proper staff.

6.4. Ad-hoc casting by skills during surgery

- During an on-going operation, a new critical condition occurs, requiring intervention of a new procedure.
- The lead surgeon uses his communicator to specify this new need.
- Each staff member in the hospital has an assigned role, which defines his/her responsibilities.
- The system sends the SMS messages to relevant skilled personnel, according to the skill-set that is predefined for the procedure.
- Presence information is used, as well as consideration of role-holder's experience.
- If a knowledge management application is available, this can be activated in parallel to deliver relevant information for the medical situation at hand. In combination with the medical information repository such as Shaman-IMR ([16]), the patient's medical history can be consulted as well. This constitutes an advanced and very desirable case of evidence-based medicine (EBM).

6.5. Nurse and other paramedic personnel allocation (dispatching)

The paramedics and nurses have very broad roles and can take on vastly different assignments throughout the day. Locating available personnel for these purposes is very important when dispatching new assignments to the most appropriate person for the job and the location. For example, these personnel may be required to roll a patient on a bed to an X-ray examination, clean a female patient, administer an injection, repair a certain medical fixture that got loose, help carry a heavy patient with a team of four strong people, and so on.

7. Preliminary Analysis of the Medical Deployment Problem

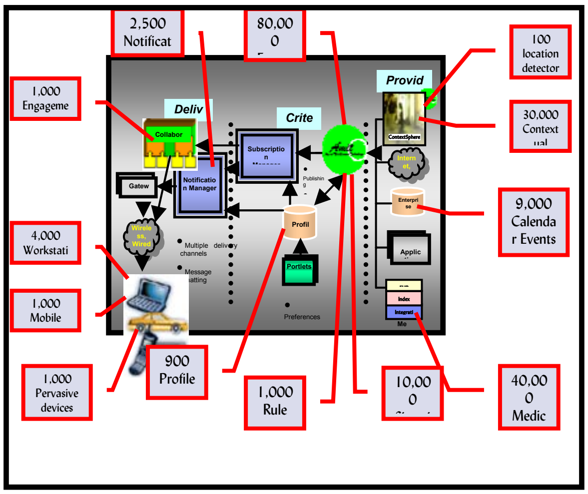

Figure 4: Situation-enriched INS means ICS, with medical environment properties.

For our purposes, we take a look at a large hospital. The following is a pretty accurate account for the size of our problem domain in this hospital, and one that we can take as a typical target site:

- 1900 beds, 1500 inpatients and 400 rehabilitation center beds, about 1500 daily occupancy
- 2000 outpatient and ambulatory visits a day
- 400 ER visits a day
- 10 different centers of expertise
- 600 medical image examinations a day
- 900 physicians, part of who are trainees
- 1600 nurses and 600 paramedical personnel (total 2200)
- 4000 workstations throughout the hospitals with full interactive and image display capabilities
- 50 meeting rooms, of which 30 are equipped with multimedia.
- 8 laboratories performing 4000 tests each day, roughly 50,000 information items per day.
- 100 consultation meetings each day
- 50 on-call consultation meetings
- 20 surgical operation theatres
- 100 surgery procedures a day

Recalling our earlier 3-part schematic of INS, we extend it in **Figure 4**. We first enrich the INS with the additional components we listed at the end of “3. The ICS architecture”. We than attach the sizing estimates based on the above data.

The rationale behind these estimates is as follows:

7.1. Providers

- 30,000 contextual events – 3100 personnel (doctors, nurses and paramedical), some 10 events per person.
- 9,000 calendar events – 900 physicians, about 10 schedules a day.
- 40,000 medical events – 40,000 lab result items and medical imaging results.
- 100 location detectors – for cellular or RF identification. The more antennas dispersed in the hospital, the better resolution, and the lower RF energy used.

7.2. Criteria

- 80,000 events – rough total of events from **Providers**.
- 900 profiles – only for physician.
- 1,000 rules – some reasonable estimates
- 10,000 situations – new events computed by AMIT's situation manager.

7.3. Delivery

- 1,000 engagements – for the 900 physicians' meeting either scheduled or ad-hoc.
- 2,500 notifications – for the 900 physicians, one per physician, and the 1600 nurses; of that, 900 are for pervasive devices, others use alternative means.
- 1,000 mobile devices – for bed-side monitoring etc., can also be providers of events.

- 4,000 workstations – for use by end users, for virtual collaboration, access to calendars and messaging sessions, etc.

8. Summary

We propose to extend the notification power of INS with a situation manager such as AMiT and collaboration capable notification technology [11] to create the Intelligent Collaboration Services (ICS). AMiT enables us to process the temporal dimension of incoming events and synthesize new events based on computed situations. Collaboration technology extends with interesting and useful possible scenarios in which ICS can manage collaboration engagements among clients, using various devices and means for physical or virtual meetings.

We see several scales of deployment agendas for ICS, but concentrate in this paper with medical domain scenarios. For this domain, we present a detailed problem sizing. The complexity metric is measured along parameters that account for volume of events and their density, and the number of clients and their interaction devices. In the medical deployment, we also propose the use of medical information for both in-organization subjects (patients) and from medical information sources such as can be obtained on WEB resources.

9. Acknowledgements

Thanks to Ayelet Biger, Ilan Schreiber, and Amnon Ribak for help during some stages of this work, and to Lisa Amitai for final editing of this manuscript.

10. References

[1] Birman, K. P. "*The process group approach to reliable distributed computing,*" Communications of the ACM 36, 12 (Dec.) 1993, pp. 36–53.

[2] Carzangia, A., Rosenblum, D. S. and Wolf, A. L., "*Design and Evaluation of a Wide-Area Event Notification Service,*" ACM Transactions on Computer Systems, Vol. 19, No. 3, August 2001, pp. 332-383.

[3] Rosenblum, D. S. and A. L. Wolf, "*A design framework for Internet-scale event observation and notification,*" ACM SIGSOFT Software Engineering Notes archive Volume 22, Issue 6 (November 1997), pp 344 - 360 .

[4] Hugues, B. and V. Cahill, "*Towards Real-time Event-based Communication in Mobile Ad Hoc Wireless Networks*," Proceedings of the 2nd International Workshop on Real-time LANs in the Internet Age. July 2003.

[5] Bennet, V. and A. Capella, "*Extending Everyplace Intelligent Notification Services to Monitor New Sources,*" March 2003, http://www-106.ibm.com/developerworks/websphere/library/techarticles/0303_bennett/bennett.html

[6] Sun Microsystems, Feb 2005, "*Java Message Service,*" http://java.sun.com/products/jms/

[7] Object Management Group, "*Notification Service Specification,*" October 2004, Version 1.1, formal/04-10-13, http://www.omg.org/docs/formal/04-10-13.pdf

[8] G. Wallis, "*MQSeries and High Availability, Part1,*" Mainframe Week, Issue 20 (May 22, 2002), http://www.mainframeweek.com/journals/articles/0020/MQSeries+and+high+availability,+part+1

[9] Carbera, L. F., Jones, M. B. and M. Theimer, "Herald: Achieving a Global Event Notification Service," http://research.microsoft.com/sn/Herald/papers/HotOS8.pdf

[10] C. Bisdikian, et. al., "*Enabling location-based applications,*" International Conference on Mobile Computing and Networking archive, Proceedings of the 1st international workshop on Mobile commerce, Rome, Italy. 2001, pp. 38 – 42.

[11] Vortman, P. and T. Hochberg, "*UCOMMIT – Unified Communication and Collaboration Model Infrastructure*," ICE, 8th Internatoinal Conference on Concurrent Enterprising, Rome, June 17, 2002, http://www.ice2002.org/day1_afternoon_session.html

[12] Luckham, D.C. and B. Frasca, "*Complex Event Processing in Distributed Systems*," Stanford University Technical Report CSL-TR-98-754, March 1998, 28 pages.

[13] Adi, A. and O. Etzion: "*Amit - the situation manager,*" VLDB Journal. 13(2): 177-203 (2004), http://portal.acm.org/citation.cfm?id=988145.988150

[14] Adi, A. and O. Etzion "*The Situation Manager Rule Language*". Proceeding of the International Workshop on Rule Markup Languages for Business Rules on the Semantic Web, 14 June 2000, Sardinia, Italy , pp. 36-57.

[15] Uri Shani, "*Towards a Comprehensive Taxonomy for Computer-Assisted Collaboration*," IBM Research Technical Report H-0220, 2004, http://domino.watson.ibm.com/library/cyberdig.nsf/papers/DE18C123B342EA1385256E7D004E1044/$File/h-0220.pdf

[16] Amnon Shabo, "*CDA in the SHAMAN-IMR Middleware from IBM Research*," (Abstract only), HL7 International CDA Conference October 7-9, 2002, Berlin, Germany, http://www.hl7.de/cda2002/absbiopres/shabo.html