A Mechanism for Managing and Distributing Information and Queries in a Smart Space Environment

Sergey Boldyrev, Ian Oliver and Jukka Honkola

Nokia Research, Itmerenkatu 11-13, 00180 Helsinki, Finland

Abstract. Distributed information management, especially when related to mobile devices with their particular connectivity and computing needs is complex. Based on notions of stability of information respository and factors such as computation resources and connectivity we can optimise the distribution and querying of said information across multiple mobile and other distributed devices.

1 Introduction

The modern communications era has brought about a tremendous expansion of wireless networks. Various types of networking technologies have been and are being developed resulting in unprecedented and increasing expansion of computer networks, television networks, telephony networks, and other communications networks. As new networking technologies evolve, consumer demand continues to fuel increased innovation with respect to utilization of networks. Wireless and mobile networking technologies continue to address related consumer demands, while providing more flexibility and immediacy of information transfer [1–4].

As the flexibility and functionality of mobile communications devices increases, options for networking technologies continue to evolve. For example, the technology associated with dynamic distributed device networks or dynamic architecture networks, such as smart spaces, are becoming increasingly practical due to the evolution of mobile communications devices.

A smart space may be an environment where a number of devices may use a shared view of resources and services to access information within the environment [10]. In this regard, smart spaces can provide improved user experiences by allowing users to flexibly introduce new devices and access most or all of the information available in the dynamic multiple device environment from any of the devices within the environment.

However, problems with information management and message routing in smart spaces can arise because smart spaces do not have a static network topology. As a result, difficulties arise with determining the location of desired information and/or the presence of intended recipients in the network. Further, the prediction of efficient paths for routing messages, such as requests for information in the smart space can also be problematic since the connectivity and the topology within the smart space cannot be assured.

In this paper we address the issue of distributing information and queries over that information and specifically the mechanisms that we have employed within our infrastructure for realising this.

Boldyrev S., Oliver I. and Honkola J. (2009).
A Mechanism for Managing and Distributing Information and Queries in a Smart Space Environment.
In *Proceedings of the Joint Workshop on Advanced Technologies and Techniques for Enterprise Information Systems*, pages 145-153
DOI: 10.5220/0002193101450153

2 Background

The Smart Space infrastructure shown in figure 1 shows our implemented architecture. The key points are that of the node, smart space and Semantic Information Brokers (SIB).

There exist numerous, individual *Smart Spaces* in which persons, groups of persons etc can place, share, interact and manipulate webs of information [11, 12] with their own locally agreed semantics. A review of such systems can be found in [13]. To interact with and apply reasoning to these Smart Spaces and there information contained therein we are required to construct processes which have access to that information - typically these are known as nodes - we also use the term agent interchangably depending on the context, but in this discussion we stick with the term ‘node’. Nodes are either personal in that they perform tasks either directly decided upon by the user or autonomously for or on behalf of the user, monitor particular situations or reason/data-mine the existing information.

A space is represented by a number of Semantic Information Brokers each containing query and connectivity engines, a store for the information - we have standardised on the use of RDF for this and a deductive closure mechanism over the store. Information is distributed over the SIBs according to local policy (described in this paper) but any node connecting (via a SIB) to the space sees only the total of all information contained withini the space, ie: if a space is constructed out of a set of SIBs B, Δ being the deductive closure over the space and $i(b)$ being the information contained in each SIB then an node will see the information as described in the equation below:

$$\bigcup_{\beta \in B} \Delta(i(\beta))$$

This implies that the SIBs are totally routable in order that all information is accessable via any SIB. Additionally within a space, even those the deductive closures are computed locally to a SIB the set of rules for deductive closure is the same for all SIBs within a given space.

Nodes access information through the SIB to the Smart Space. The SIB operates as a single entity in managing a Smart Space. However, as it stated in [5–7], the infrastructure can be distributed. The SIB can be formed by set of more than one SIB which represents the Smart Space.

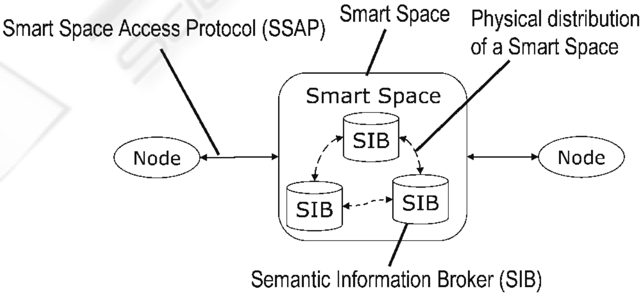
Smart Space Access Protocol (SSAP)
Smart Space
Physical distribution of a Smart Space
Smart Space
SIB
Node
Node
SIB
SIB
Semantic Information Broker (SIB)

Fig. 1. Smart Space SIB/Space/Node Architecture.

This is done in a way that all SIB can receive all messages arriving at any SIB and all messages passed on by the SIB appear to come from a Smart Space. Infrastructure presupposes communication between any of SIBs which form one single Smart Space. Such communication is performed by means of the adaptive designation of a SIB "master" in any given Smart Space, throughout the Smart Space evolution as shown in figure 2. To leverage the abstraction of SIBs addressing, the IDs (keys), are used to identify each message session to determine whether a SIB should process a particular message of not. These keys are used to identify which SIB is in charge for processing which message type and is also used for the workload balancing over all SIBs.

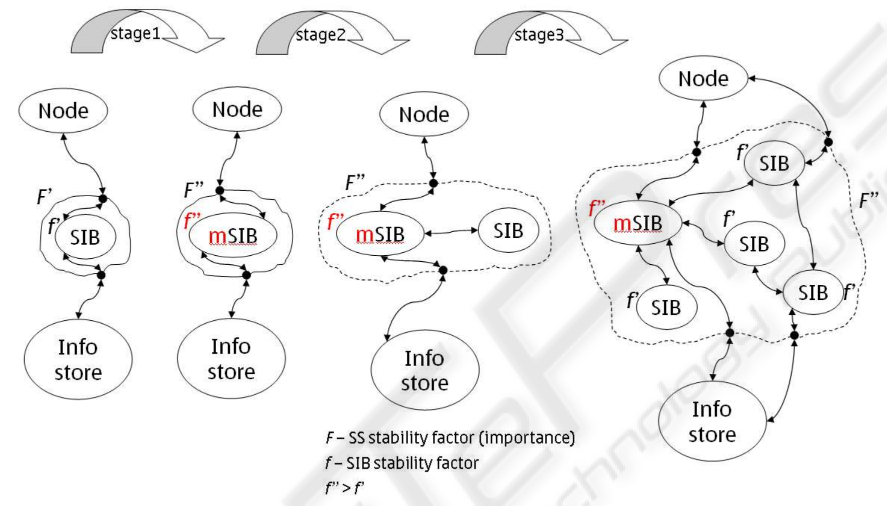

Fig. 2. Smart Space Evolution.

This solves the problem of distributing information and queries in a consistent manner across multiple computation elements making up the information processing system of a SmartSpace. Furthermore, the issue of computing the deductive closure of information at run-time is possible. Marking in terms of priority and stability level of the individual processing elements such that queries can be distributed such that the most efficient processing of the query is made with the most stable set of information.

Stability factor is constructed from the connectivity controller information, where the SIB is running, any additional valuable environment information. As an example the Stability factor can be represented in folowing form:

$$S_j = C + Idiv_j + PRL_i$$

where *S* is the stability factor for *j*-th SIB entity of any *i*-th device, *C* is connectivity weight provided by connectivity controller, *Idiv* indicates how information is diverging in the sense of query satisfaction, and *PRL* is optional cumulative power/reliability/latency function provided by the particular device.

As it is shown the Stability factor is a result of superposition between information which is provided by the intelligent connectivity controller, e.g. physical means which form computing environment, and Information management infrastructure [9]. Intelligent connectivity controller is aware of its surrounding environment in addition to its local connectivity capabilities. It may also be aware of the physical limitations and of individual devices within the wireless range or [11] within the wired network is associated. The aim of the interface provided by the controller to the distributed framework component is to provide enough yet abstract connectivity properties of the devices participating in to the distributed network. In addition the controller hides the complexity of multitransport control mechanisms as a connectivity cost function interface.

Relationship between Stability factor and the Strategies shapes any stable (reliable) distribution between several SIB entities, e.g. distributed SIB. The Stability factor is defined in tight connection with the strategies that form the distributed solution feasible.

Several strategies are considered:

1. to distribute query around all instrartucture, and, to put insert/retract on any stable SIB
2. to distribute inserts/retracts around all infrastructure, and, to put query on any stable SIB
3. any balancing strategy between listed above

SIB is stable if its operational parameters produce Stability factor as above a certain threshold, thus room for real-time optimisation exists.

A Smart Space is actually constructed from a number (at least one) SIB which contain the aforementioned schedulers, management information, listeners and information store. A Space is represented by these SIBs: the total possible connectivity methods is given by the distributed union of all the representing SIB listeners.

SIBs communicate internally to ensure that membership and credentials of the Nodes that have joined that Space - they may connect via any listener and any SIB in that Space. From the point of view of any Node, the information available is always the distributed union over the transititve closure of the routes between all the SIBs - the SIBs contain routing tables to other SIBs and within a Smart Space all the SIBs are totally routable but not necessarily totally connected.

3 Considerations

We take into consideration the following:

1. Optimal distribution mechanisms and policies
2. Distributed query mechanisms
3. Optional mechanism based upon distributed inserts/retracts

There are three message groups which are used to provide distributed infrastructure: management, store (subscribe, cancel subscribe) and query (query, deliver).

The distributed infrastructure management messages are used to keep correct neighbour descriptors and incremental routing tables and are used to support store and query messages. They are the basic elements of synchronisation protocol between any of the SIBs. The incremental routing table maintained to handle a Neighbours list. There are following maintenance messages, Join/Leave, HB (heartbeat, including timeout), Send, Notify and Forward. There is a cache for previously constructed routes, thus messages can be routed directly, without lookup within the distributed infrastructure. To inform about connectivity parameters of any SIB instance the HB (Heartbeat) message is used. To navigate around the SIBs two lists are constructed and used, the first is based on Neighbours list and is in charge of any successor routing, the operation is simple Send. The second list can be used to improve overall performance and leverage the workload, that list is exponentially distributed IDs of all other SIBs.

Store group are used to pass information to the SIBs and to operate with them. They are Insert and Remove, provide distribution of the information. Query message (below) presents the way of distribution which is applicable to Store group messages.

The Query group is used to pass the query to the SIBs for any particular information as well as subscriptions to the SIBs and to operate with them. They are Subscribe and UnSubscribe, provide distribution of the persistent queries. Query message presents the way of distribution which is applicable to all Query group messages. The Query message can be routed to any other SIBs. The Query message can be forwarded by any other SIBs according to the incremental routing table, finger table and routing decision. In case of message forward, notification is generated by SIB and is used to override routing decision. Such decision is obtained by means of network conditions. Thus the query tends to pass SIBs only once. Therefore the query route construction converges to the efficient query forwarding mechanism. The result for Query group is passed by means of Deliver message, which returns particular information.

SIB has managing responsibilities to form the SIB distributed infrastructure. Such managing SIB is denoted as mSIB instance, the master SIB. There can be more than one mSIB. Each SIB should have stability factor. In negotiation of any two SIB the SIB with highest stability factor is selected to be mSIB. Stability factor is a result of superposition between information which is provided by the intelligent connectivity controller, e.g. physical means which form computing environment. Intelligent connectivity controller is aware of its surrounding environment in addition to its local connectivity capabilities. It may also be aware of the physical limitations and of individual devices within the wireless range or within the wired network is associated. The aim of the interface provided by the controller to the distributed framework component is to provide enough yet abstract connectivity properties of the devices participating in to the distributed network. In addition the controller hides the complexity of multitrasport control mechanisms as a connectivity cost function interface.

In a very beginning, a SIB or the SIBs should establish the SIB distributed infrastructure by sending a Join message as shown in figure 3.

Such Join message contains an authentication part, [1] When the mSIB receives this Join message it checks it with authentication mechanism [1] and if it fails it tells the newcomer that Join has failed. If the mSIB does succeed with newcomer authentication it sends confirmation to Join message. When the confirmation to Join message received then the SIB is confirmed SIB distributed infrastructure member.

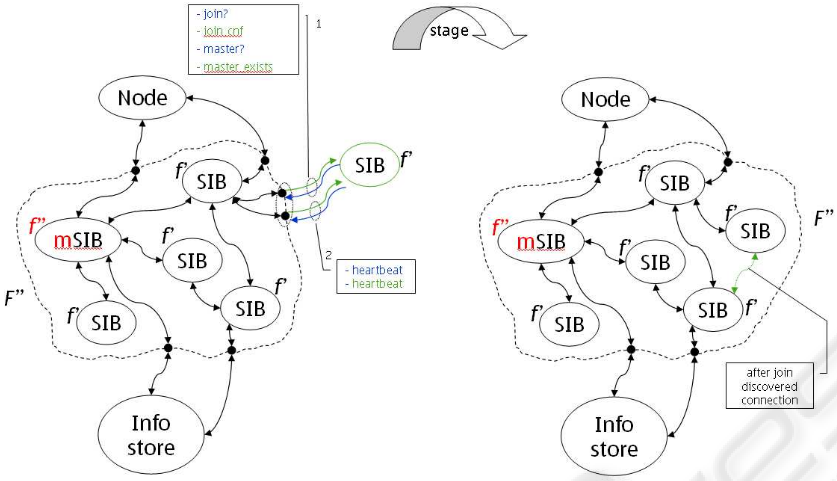

Fig. 3. Smart Space, join procedure.

4 Description

Since the SIB distributed infrastructure management is driven by the Heartbeat message exchange, each SIB has timer to track these Heartbeat messages. The events whether the SIB received Heartbeat message from mSIB or any neighbour SIB within a certain period of time are tracked.

Regarding mSIB to SIB interaction, there are several cases when the SIB should execute Join the SIB:

- If the timer expires, no Heartbeat message is received in a given period of time will mean that the mSIB is down and the SIB should Join the SIB distributed infrastructure again.

- If it gets Quit message, SIB should leave the SIB distributed infrastructure

If the SIB sending Join message doesn't have a confirmation message this SIB broadcasts messages offering to become the mSIB. If the SIB gets “mSIB exists” message this SIB tries to Join again with preserved stability factor. If after the SIB broadcasts the packets offering to become the mSIB, it gets no response within predetermined amount of time, it sends broadcast responses to inform everyone in SIB distributed infrastructure about new credentials and from now on acts as the mSIB of SIB distributed infrastructure. If there is response from mSIB, and if the stability factor is above the threshold to become mSIB, it sends broadcast responses to inform everyone in SIB distributed infrastructure about new credentials and from now on acts as the mSIB of SIB distributed infrastructure.

If during the process of mSIB handover as shown in figure 4, the SIB has no notification that another mSIB exists and is thinking that it is the only mSIB but keep receiving Quit message, the SIB must return and try to Join again.

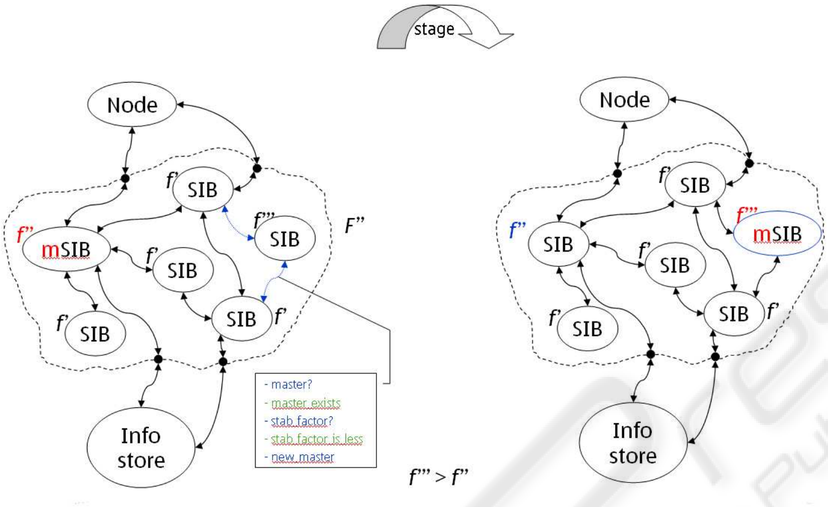

Fig. 4. Smart Space, handover procedure.

During the lifetime due to the fluctuations of the stability factor of any SIB, including current mSIBs, and, due to involuntary faults, the managing functionality can migrate to any other SIB.

There are three main approaches to maintain the infrastructure (chained by stability of the distributed infrastructure, it means that each stage represents maturity of the infrastructure and is used by controllable takeover mechanism):

1. all Smart Space members, SIBs, are able to see all packets and applies incremental routing technique to forward the message to the appropriate Smart Space member, SIB
2. mSIBs are managing and determining the packets to be send and forward to the appropriate Smart Space member, SIB
3. when SIB routing mechanism has query update capabilities, then this method is used. It is the case when routing technique is extended by the local information that particular SIB can provide [9]

The mechanism of determination which Smart Space member is to process a message and allow other to disregard the message is based on incremental key routing [6].

5 Discussion and Conclusions

Our approach intelligently utilises stability factor calculation based on the information management approach [9]. Therefore, it serves as leveraging information dissemination throughout the balanced query/insert/retracts. Tracking the most relevant requests of the information through the scalable, energy efficient mechanism. Whole approach efficiently utilises different computing platform, though some complexity in computations exists.

This approach can also be applied to improve the fault-tolerance of the space as a whole to information loss. While not our targetting focus this particular area provides many suitable ideas which are being explored.

Acknowledgements

This work has been partially funded as part of the TEKES ICT SHOK DIEM (www.diem.fi) project.

References

1. Rajeev Gupta , Krithi Ramamritham, Optimized query planning of continuous aggregation queries in dynamic data dissemination networks, Proceedings of the 16th international conference on World Wide Web, May 08-12, 2007, Banff, Alberta, Canada
2. Chris Olston , Jing Jiang , Jennifer Widom, Adaptive filters for continuous queries over distributed data streams, Proceedings of the 2003 ACM SIGMOD international conference on Management of data, June 09-12, 2003, San Diego, California
3. K. Zhang , H. Andrade , L. Raschid , A. Sussman, Query planning for the grid: adapting to dynamic resource availability, Proceedings of the Fifth IEEE International Symposium on Cluster Computing and the Grid (CCGrid'05) - Volume 2, p.751-758, May 09-12, 2005
4. Luis A J,et al. ARIADNE: A system for constructing mediators for Internet sources (system demonstration). In:Proc. of ACM SIGMOD Conf. on Management of Data,New York:ACM Press,1998. 561-563
5. Ian Oliver, Jukka Honkola, Personal Semantic Web Through A Space Based Computing Environment, In proceedings: Middleware for the Semantic Web, Seconds IEEE Interntional Conference on Semantic Computing, Santa Clara, CA, USA, August 4-7, 2008
6. S. Boldyrev, S. Balandin, Illustration of the Intelligent Workload Balancing Principle in Distributed Data Storage Systems. Proceedings of workshop program of the 10th International Conference on Ubiquitous Computing, September 2008
7. A. Lappetelainen, Juha-Matti Tuopola, Arto Palin, Timo Eriksson, Networked systems, services and information, The ultimate digital convergence, 1st International NoTA conference, Helsinki, Finland, 2008
8. Niki Trigoni, Yong Yao, Alan Demers, Johannes Gehrke, and Rajmohan Rajaraman. Multi-Query Optimization for Sensor Networks.. International Conference on Distributed Computing in Sensor Systems (DCOSS 2005).
9. S. Boldyrev, I. Oliver, R. Brown, J.-M. Tuupola, A. Palin, and A. Lappetelainen, Network and content aware information management, submitted and awaiting review.
10. Ian Oliver and Jukka Honkola and Jurgen Ziegler, Dynamic, Localised Space Based Semantic Webs, WWW/Internet Conference, Freiburg, Germany, October 2008

11. Reto Krummenacher, The Smartest Space of All: A Global Space of (Machine-Understandable) Knowledge, 1st Russian Conference on SmartSpaces, ruSmart 2008, St.Petersburg, Russia, September 2008.
12. D Khushraj and O Lassila and T Finin, sTuples: semantic tuple spaces In: Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. 22-26 Aug. 2004
13. Lyndon J. B. Nixon and Elena Simperl and Reto Krummenacher and Francisco Martin-Recuerda, Tuplespace-based computing for the Semantic Web: a survey of the state-of-the-art, The Knowledge Engineering Review, June 2008, 23:2 pp181-212