Architecture for Application Development on the fly for Wireless Sensor Networks

Poonam Yadav
Department of Intelligent Systems
Indian Institute of Information Technology
Allahabad, India
Email: yadav_poonam@is.iiita.ac.in

Nagesh Yadav
Department of Intelligent Systems
Indian Institute of Information Technology
Allahabad, India
Email: nagesh@iiita.ac.in

Anuragmayi Thuremella
CIMIC, RUTGERS University
Newark, NJ, 07102
01-973-353-1865
Email: anurag@cimic.rutgers.edu

***Abstract*— Wireless sensor networks are designed and built as per the application's needs and thus cannot be shared by other applications. This makes the deployment of applications expensive and the applications are always built after the initial event. The authors propose and explain the novel concept of application development on the fly and propose the architecture for such a system. They also propose a strategy for the reuse of the elements of the pre-existing application networks, by the new application under development. The application development in real time when the event is taking place, assumes the availability of the base knowledge needed for the network somewhere on the net or the availability of human expert. The proposed architecture uses the following concepts: semantic interpretation of low level signals reaching an application server, identification of the problem, problem decomposition on the fly, identifying SW agents that could solve sub problems and integrate the sub problem solutions into the complete solution. It makes use of distributed knowledgebases across the net and data-mining techniques. The paper identifies the major challenges in developing such systems.**

I. Introduction

As and when the society feels the need, we identify an application to meet the need and then design and deploy a suitable sensor network. Let us take an example of sensor network that belongs to the Indian Meteorological department for weather monitoring and forecasting. Some other organization such as a university might also like to obtain similar data from the same region, but for some other application or some very similar application. They may have to set up their own sensor network since the existing network cannot meet their needs. Thus every different and newly identified application needs a different network. We might think of replacing the conventional sensors with wireless sensors for the obvious reasons and yet each application would be having its own sensor network. In each case, the sensors belonging to a particular network would send the data to the corresponding application system and this triggers the desired application. Thus it is easy to visualize the sensor data triggering the pre-identified applications. The scenario would remain the same even if we replace our older telemetric sensors by wireless sensors. Building and deploying such networks by each organization for specific applications would be prohibitively expensive and would act as a deterrent for the deployment of new applications. It makes sense if the sensors deployed by an organization for it's own application could be used by some other organization for a different application, with both the applications running simultaneously. Thus sensors deployed by either of the applications are shared by the other application. In such a case the sensors trigger either of the applications. However, consider the case that an event like Tsunami takes place and is detected by the sensors and they in turn send the information to the the application server. Let us assume that the application server does not have pre-defined application for Tsunami and so cannot respond to the Tsunami signal. However a human being placed in the position of the application server would be able to build up an application or solution as the events unfold. Thus there is a need for a system that would emulate a human being to provide solutions to the problems that are previously defined and as well to the problems that are yet to be defined. This would involve identification of a problem from the sensor data and then find a solution for the undefined problem. As an example consider the case of environment monitoring discussed in the next paragraph.

For weather forecasting let us consider that we have deployed a number of sensors for measuring the parameters like temperature, pressure, humidity, rainfall etc. These sensors provide the data for standard applications. However when data provided goes beyond the range of normal data then there is the need for intelligent problem identifier, and solution provider that can advise the user accordingly as the standard applications are not meaningful in such situations. There is also the requirement for systems that could manage the queries/requests coming from the user and provide appropriate solution without the user intervention. This application or system may further work in a hierarchical manner to decompose the problem and integrate the solutions from bottom to top. Different levels in hierarachy may work concurrently, parallely or sequentially, depending upon the problem that has to be solved. Thus the need arises for SW modules for event detector, event identifier, Problem Decomposer and Problem solver on the fly, and all these units make use of the knowledge available on the web, more precisely, the semantic web. Further the system works in a dynamic environment.

In the application development, there arises the need for the fusion of sensor data, and integration of sensors, processing applications, and actuators. The whole application development architecture is build around the knowledgebases and the information must be available on the web. We might consider that the human being is a natural extention of the knowledgebase of the semantic web and also the extention of the concept of agent. In this scenario the human operator is fully integrated into the world of sensor network.

A. Issuues and Challenges

The main challenge in the development of applications, on the fly is the heterogeneity of knowledgebase's and the retrieval of meaningful information from them. We assume that somewhere on the web, the required knowledge exists in some form or the other such as documents or databases or knowledgebases or human users providing assistance through some interface and the desired information has to be accessed from these sources in an automated manner. In wireless sensor networks, there is always the requirement of communication and collaboration among sensor nodes, and the data is distributed over the entire heterogeneous networks. A general Problem solver architecture is required to solve all such problems.

II. Architecture of Flexible Information Systems

Keeping these requirements in view, the authors suggest a flexible and novel information system architecture that would be useful for the application development on the fly for wireless sensor networks and for other applications. The architecture should segregate applications from the data/information and the data is often in the form of either databases or knowledgebases or expert systems and the heuristics link the applications to the relevant information/data. The Architecture proposed is given in figure 1 below:

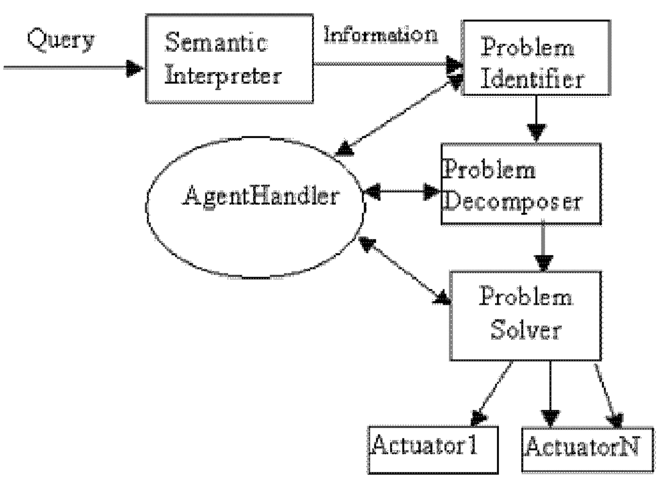

Fig. 1. Architecture of flexible information system

The proposed architecture built around the agent technologies, would not only use the existing databases/ knowledgebases/information, but would also use future databases even when the existence of such databases is not explicitly known to the application system. The application systems are built on the fly using semantic web concepts[2]. Suggested architecture makes use of the distributed multiple agent framework paradigms[1]. Agent may be seen as a software program that perceives its environment through either HW or SW sensors and acts upon the environment with a certain degree of autonomy, while performing its task. Agents are often assigned tasks to be performed by the human beings and they (the agents) in turn would pursue the tasks autonomously and would report the results to the system that deployed them.

The working of the system is explained below:

1) The user's request comes through graphic or voice or text format or data from an interfaced sensor reaches the semantic interpreter. The semantic interpreter using its own knowledge base and data mining techniques would interpret the requirement and would ask the user to confirm its diagnosis of the requirement, where such confirmation is needed.
2) After the user confirms, the problem is identified and is sent to Problem/Application decomposer.
3) The problem/application decomposer would check whether such an application already exists using its agent, and if so, it would locate the relevant application and database/knowledgebase and would run the application.
4) However if the application does not exist, then it would decompose the problem into component problems, and these are passed on to the AgentHandler (See figure 1)
5) The AgentHandler has following components:
    a) The facilitator agency would use its agents and search agent registration sites (a Component of knowledge Provider) and locate suitable agents that could perform the component tasks. The list of the agents that could perform each of the tasks is the outcome from the facilitator agency. It is assumed that all the agents that are created by different persons/systems and are on the net, register their location and their capabilities at certain sites called Agent Registration Sites(See figure 2 and 3). The agent(s) of the facilitator agency search these Agent Registration Sites and retrieve lists of agents for each task. These lists are passed to the Agent Selector.
    b) The Agent Selector prepares the requirements of subtasks and sends them to the lists of agents soliciting them to inform whether they can do the task. The requirements are detailed performance specifications for each task including the inputs and outputs and their formats, the resources needed to perform the task and the time frame of execution of the task. Agent responses are received by the Agent selector and it selects suitable agents for solving the problem. A standby agents list is also prepared. This information is sent to the Problem Solver.

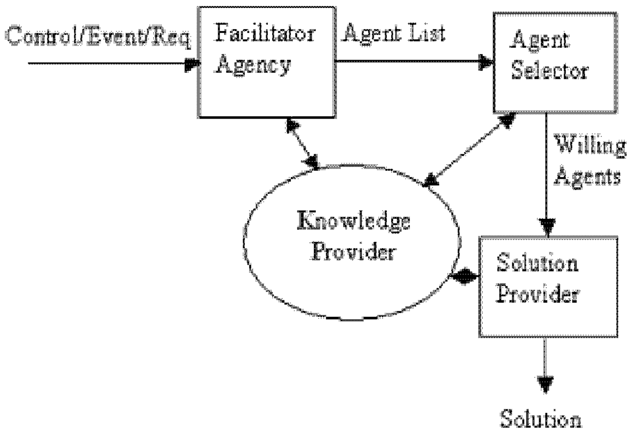

Fig. 2. Components of AgentHandler

c) The Problem Solver assigns the tasks to the selected agents and integrates the solution as the results come. Final solution is presented by the problem solver.
d) Knowledge Provider has three components: Domain Knowledge, Registry scouting and Discovery Services[5]including agent's registration sites and Semantic web services.

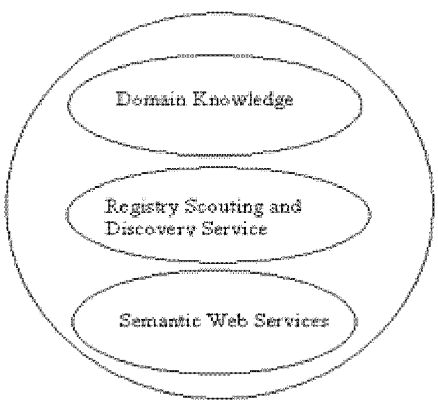

Fig. 3. Components of knowledge Provider[5]

i) Domain Knowledge: Domain knowledge will be associated with all the four key components of Information system (see fig 4.)[6]. Domain knowledge for semantic Interpreter also includes Ontologies for query processing[7]. This will handle schema, metadata for approximating queries. Similarly domain knowledge for problem identifier, Problem decomposer and Problem solver include all the appropriate algorithms required for particular set of problems. There is learning mechanism that can automatically update this knowledge base. Domain knowledge makes use of the already existing documents and databases. The extraction of information from documents(that could be text or image or data files etc.) and databases is by either rule based expert systems or the database accessing application. Once the meta information is extracted, software agents could make use of that information for solving problem. This enhances the reusability concept of the already existing databases and documents.
ii) Registry Scouting and Discovery Services: Scouters[5] are registries and discovery services that help applications find the right agents who can do the job efficiently and in an optimized manner. Registry Mode provides a mechanism for different agents to add itself into the system by advertising its capabilities and registering into appropriate registries. At this time, each agent has the link of all the semantic web services for a particular situation[3]. Agents now launch concurrently to find the solution from different semantic web services.
iii) Semantic Web Services: The concept of semantic web[8] enables the development of information space, and the language for expressing information in a machine processable form, which in turn enables the intelligent software agents to interpret the semantic web.

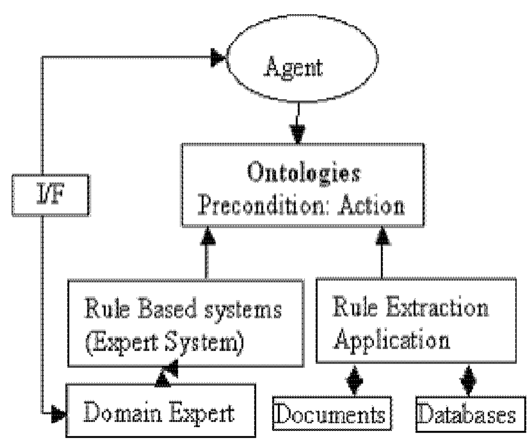

Fig. 4. Interaction of Agent with Existing Web

III. Conclusion

This distributed agent architecture allows the construction of systems that are more flexible and adaptable than distributed object framework[6]. Human being could be viewed as an agent and in such a scenario, the human user should be able to interact with the collection of distributed agents as an equal member of the system, and not just an outsider to whom the results are presented. The architecture offers enormous capabilities to the system. Of course, the implementation of such a system is a challenging task but not beyond the reach of the current technology.

ACKNOWLEDGMENT

The authors would like to thank and acknowledge Prof.M. Radhakrishna of IIIT, Allahabad for the help and contribution in enhancing the ideas. We also thank IIIT, Allahabad for providing excellent lab facilities and motivating research environment.

REFERENCES

[1] Jiang, Guafei, Chung, Wayne W., Cybenko, and George, "Semantic Agent Technologies for Tactical Sensor Network," In Proceedings of the SPIE, Volume 5090, pp. 311-320 (2003).
[2] Poonam Yadav, "Information Access by Rural Population -A novel architecture for information Systems," National seminar on Empowering Rural India through ICT, in IETE in April -2006.
[3] Daniela Rus, George Robert Gray and David Kotz, "Autonomous and Adaptive agents that Gather Information," In AAAI'96 International Workshop on Intelligent Adaptive Agents. Proceedings available as AAAI technical report WS-96-04.
[4] Adam Cheyer and David Martin,"The Open agent Architecture,"Autonomous Agents and Multi-Agent System Journal, Volume 4, Issue 1-2 (March-June 2001), Pages: 143 - 148.
[5] Anuragmayi Thuremella, Aabhas Paliwal, Poonam Yadav and M. Radhakrishna, "DOMA: Deploy Once Multi Application Sensor Network Framework," Proceedings of the International Conference on WCSN-2006, Dec,2006.
[6] Ana C. Munoz and Jose L. Aguilar, "Architecture for an Intelligent Distributed Database," Proceedings of the 13th Euromicro Conference on Parallel Distributed and Network(2005),Pages: 322 - 328.
[7] Olivier Corby, Rose Dieng-Kuntz, Catherine Faron-Zucker, Fabien Gandon, "Searching the Semantic Web: Approximate Query Processing Based on Ontologies," IEEE Intelligent Systems, vol. 21, no. 1, pp. 20-27, Jan/Feb, 2006.
[8] T. Berners-Lee, J.Handler, and O.Lassila. " The Semantic Web," Scientific American, May, 2001.