A Distributed Model for Approximate Service Provisioning in Internet of Things

Chayan Sarkar
Delft University of Technology
Netherlands
C.Sarkar@tudelft.nl

Vijay S. Rao
Delft University of Technology
Netherlands
V.Rao@tudelft.nl

R Venkatesha Prasad
Delft University of Technology
Netherlands
rvprasad@acm.org

Abdur Rahim
CREATE-NET
Italy
abdur.rahim@create-net.org

Ignas Niemegeers
Delft University of Technology
Netherlands
I.G.M.M.Niemegeers@tudelft.nl

ABSTRACT

With the advent of Internet of Things (IoT), many newer possibilities of service provisioning are opening up. Truly Google like searching for objects in our daily life is not ruled out in near future. Towards this aim, we have proposed *Approximate service provisioning* in our previous works [18, 19]. In this paper we propose a model for such a service. We discuss briefly the notion and concept of approximate services first and then we elaborate on the ideas of achieving such a paradigm. Further, since we know that there could be enormous number of devices in the future and in particular in IoT, a centralized solution seems impractical. Thus, we throw some light on the use of distributed approximate service models in this article.

Categories and Subject Descriptors

C.2.4 [**COMPUTER-COMMUNICATION NETWORKS**]: Distributed Systems

General Terms

Design, Internet of Things, Service provisioning

Keywords

Approximate Service, Service Daemon, Service Proxy Proceedings

1. INTRODUCTION

It is predicted that there would be 7 Trillion devices for 7 Billion people by the end of this decade [14]. This is to an extent plausible in the wake of newer communication technologies rolling out day by day. Thus each device has some capability of identifying itself in the digital domain and communicate with the rest of the world (at least partially or even may be one-way). For example, an RFID tag could identify a box of fruits and could be traced back in digital domain. One can exactly find its origin and when it could be perishable, or what sort of treatments it requires so that it is useful in the near future. Further, this information helps in tracking, managing, and supply chain management too. On a similar note that it is also observed that Internet of Things (IoT) paradigm is catching up with reality. Architectures and framework for IoT has been studied by industries and academia under various projects [1, 4, 5].

With miniaturization and high communicability of newer ICT (Information and Communication Technology) devices, any physical object in this world could be, in principle, made to talk to any other object in this world from anywhere. Further actuators could help in taking actions and also controlling the way *the things* behave in the environment by passing on the control signal to them. Thus intelligent applications can be built around IoT making the surrounding environment much more comfortable to live while making the best use of resources and facilities available in the world – energy, food, transportation, entertainment, etc. We should also mention here that it is not always humans involved in this new wave of applications, device-to-device communication – popularly known as Machine-to-machine (M2M) communication – is also a major player. Economic feasibility in the changed world order (indeed affordability by many strata of societies) is also fueling this wave of newer applications.

With this background, a deliberation on the number of applications in this new world of IoT reveals unimaginable possibilities. With everything surrounding us being communicable, as an example, we could make a Google type of search application to find what we want. The true innovation would be that, even when we do not have complete clarity of what exactly we are looking for, we are able to search for something of similar character (often we do web search even though we do not know exactly what we are looking for). This requires a massive number of devices around us, and intelligence. We argue that with the advent of IoT, we have indeed many possibilities for such an innovation. We also believe that intelligence could be built into the devices however miniature they may be. Thus, we think that the next step would be to come up with a new paradigm where

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copiesbear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Self-IoT '12 September 17 2012, San Jose, CA, USA.
Copyright 2012 ACM 978-1-4503-1753-5 ...$15.00.

we can think and act approximately towards a service that is not exact but in the same direction as one sought for. We had shown earlier that an *Approximate Service* concept is indeed plausible in our earlier works [18, 19]. For example, if a user searches for a hammer to nail a picture into a wall, an approximate service search will give a stone as shown in Figure 1. The system needs to be intelligent enough to point you to a small stone for this application, rather than a rock.

Another line of reasoning is that the increased dependency on (networked) devices for daily living will increase our expectations about the availability of services. In real world, we notice that a perfectly matching service for a requirement (or tuned to a situation) is not always available. In these situations, if an approximate and an alternative service for the required one is available, then that can solve the immediate necessity. The idea is to extend this notion to the services offered by the ICT world. This paper dwells deep into this new conceptual paradigm and proposes next steps in building it. We note here that it is not possible to provide a centralized, highly intelligent solutions. Since the number of devices (and actuators) are enormous so that it is impractical to aggregate all the information in one place and take action centrally. Thus, we propose a distributed approach for provisioning approximate services.

Figure 1: You searched for a hammer; you got a stone

The rest of the paper is structured as following: Section 2 provides a brief note on the related explorations underway. Section 3 discusses how each and every device is represented in a digital world so that they can be located by a searching mechanism. Section 4 describes how a service can be located using a service lookup model. Finally we conclude the paper along with a list of future work in 5.

2. RELATED WORKS

NSF and EU have funded projects to investigate Future Internet. The Future Internet initiatives, by NSF, in Future Internet Design (FIND) [2] investigates designing future networks that are more secure and available than today's Internet or by ensuring that functions like information dissemination, location management or identity management fitting in new design and environment. FIND also investigates how economics and technology can interact to shape the overall design of a future network. The NSF is also taking further steps to address the scalability of the future Internet by various means. One such attempt where many ideas by leading researchers were shared and discussed can be found in [7].

Many European FP7 projects, such as FIRE, FIND, GENI, etc also investigate Future Internet paradigm in great details. They describe the application of concepts like large scale networking, Cognitive Networking (including Cognitive Radios), network of networks, as well as architectures developed for a converged communication and infrastructure services. In the EU project MAGNET and MAGNET-Beyond, the concept of Personal Networks [13] has proposed, which is similar to Internet of Things. Personal Network Federations [11] proposes many devices of several people cooperating to achieve several tasks. In these projects, several service architectures have been proposed [12, 15]. The European commission has taken a big step in identifying issues and encouraging new and innovative ideas under FIRE initiative [3].

There are other EU FP7 projects that investigate the provisioning of Internet of Things like IoT-A [5] and IoT.est [6]. IoT-A proposes to create of an architectural reference model while defining an initial set of key building blocks, thus proposing to lay a solid foundation for future IoT. IoT.est project envisions to create an effective dynamic service creation architecture. The basis for this the popular Service Oriented Architecture (SOA) [8]. SOA does not discuss dynamic service creation, while IoT.est aims at provisioning them. In IoT.est, the service creation is divided into two phases: (a) design time phase where modeling and composition of services take place; and (b) run time phase where service provisioning happens over two sub-phases - deployment and execution.

One could also find the Approximate Services concept towards the vision of pervasive Future Internet which is invisible but always present to support users [18, 19]. Further, there are already some work is currently in progress under EU FP7 project iCore [9]. Approximate services certainly needs dynamic service provisioning architecture along with dynamic functional abstraction and classification. We now proceed to explain one possible way of realizing the dream of Approximate Service and related issues in the sequel.

3. REPRESENTING OBJECTS IN THE DIGITAL WORLD

When hundreds of thousand devices are connected together in way that they can act smartly by exchanging information among themselves, they form *Internet of Things*. Today what we see as Internet is not a centralized system. Rather, any device following the TCP/IP protocol stack can be connected via Internet. But when we talk about Internet of Things, we have to tackle heterogeneity among objects to make them connected. Also machine to machine interaction need to be increased to establish an Internet of Things. To tackle technical heterogeneity, the physical communication needs to be abstracted from the application and each ob-

ject needs to be represented in such a way that they can be searched and communicated easily.

3.1 Objects and Virtual Objects

We envision that any physical entity could be represented in the digital world. It is very easy to see that a device with ICT capability could be represented by its representative *object* in the digital domain, i.e. its interfaces, capability sets, its functionalities and the services it may offer. For example, an object representing a printer with its own IP address could be stored in a server with many of these details and it could be accessed and addressed whenever required. However, with the advent of Internet of Things and massive ICT infrastructure, it should also be possible to represent real world things as an object in some form and stored wherever possible. For example, a table, a chair or a coffee cup or even an apple. These representations we call it as a *virtual object*. These objects could be accessed in an opportunistic way to provide what we call approximate services through some form of sensing which may not provide required information exactly.

3.2 Sensing

An object, without any communication capability cannot be controlled or interfaced with any ICT means. Though now a days more objects are equipped with communication capabilities (may be a Bluetooth or a simple RFID interface), many objects with its virtual representation will not have means to communicate by themselves. If any of these objects are required for some service, locating the required object need to be done with the help of another device which is ICT enabled. A number of sensors with various capabilities can be deployed in the interested regions. Thus either by direct communication or with the help of these sensors, an object's location can be identified. However, the question is, how many sensors should be deployed optimally to cover maximum area or maximum number of objects. Alternative approaches to this direct sensing method can be virtual and indirect sensing.

3.2.1 Virtual Sensing

Sensing obtains data from direct measurement of the physical phenomena by placing sensor (transducer) physically at the location or on the object of interest. However, in many real life scenarios, obtaining the sensing data by placing the sensors at the exact location is difficult or even impossible. An alternative sensing method, know as "virtual sensing", produces sensing data from other sensors which may not be even present at the location of interest. *Virtual sensing*, which uses physical sensing data and a suitable model to grasp the information of interest that is difficult or impossible to reach [16]. Depending on the model (used to produce the sensing data for the virtual location) there is always a chance of inaccuracy as there is some kind of approximation involved. In the context of service approximation, some form of virtual sensing is needed. In our previous example of coffee machine, locating a glass for drinking water would be difficult if the glass (in fact most of the glasses) does not have some form of communication apparatus. The location can be identified by virtual sensing with the help of the coffee machine. An operational coffee machine can indicate the availability of glasses.

3.2.2 Indirect Sensing

Another form of approximate sensing is indirect sensing. The idea is to avoid the large scale direct and dedicated deployment of sensors[20]. Sensing data captured by a sensor (with a particular purpose) can be used for another purpose. For example, suppose we have installed sensors in every room of a building which switches on light when there is someone inside a room (say with motion or IR sensors). Otherwise the lights are switched-off to avoid electricity wastage. Now this sensing data can be used by the building security to infer whether a person is inside the building or not. This could be used in many ways in different situations.

4. SERVICE LOOKUP

Finding a service or just even locating an object as explained in our example would be highly sought for objective in the Future Internet. A service can be thought of as a conglomeration of functions and each function may not be served by a single objects. The service proxy breaks a service into smaller functions. If an exact functionality cannot be provided by any objects in the vicinity, the proxy tries to find an approximate alternative. From a very high level view finding a service (or identifying an object) is composed of two simple task - (i) representation of each objects in a manner so that they can be fetched in searchable environment and (ii) a search engine with required awareness to locate a service (possibly a list of available services with some ordering). The first part is described in the previous section. In this section, we will provide a distributed model for finding a service and then we will try to visualize how a service can be found using this model.

4.1 Notion of a Service Daemon

To realize the distributed search model, we are proposing a software entity, called the *Service Daemon*. It will be running on each devices after the device is powered on (booting). When a device need to search for a service (based on a user request or an application requirement) the service daemon starts asking its neighbors for the possible service provider. To organize the tasks need to be done by a service

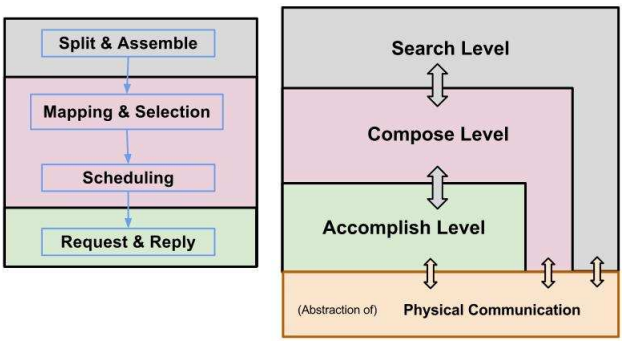

Figure 2: Levels and their responsibility in a Service Daemon

daemon, three levels are defined within it. Figure 2 shows the levels in a service daemon along with the responsibilities for each of the levels. The *search levels* of neighboring (device) service daemons communicates to gather information about possible service providers. The *compose level* makes the decision about choosing a proper service provider. In

case of a complex service request which need to be served by multiple provider, the search level breaks the job into smaller sub-tasks/functions (Figure 5). The compose level, then decides on the most appropriate service providers for each sub-tasks and finally schedule them. Choosing the appropriate provider requires cognitive ability and it is described in more details in Section 4.3. The *accomplish level* works when a device itself acts as a service provider. After scheduling the sub-tasks at the compose level (at the requesting device), the services are retrieved from the accomplish level of the providers (Figure 5).

4.2 Service Proxy: backbone of the distributed search model

Service Proxy is nothing but another device which has much higher resource availability than a simple sensor device in terms of memory, computational capability, energy etc. The service daemon also runs on the device which acts as a service proxy. When a device is not capable enough to do full fledged operations as a service daemon, i.e. splitting and searching for a service provider, compose the service out of multiple sub-tasks etc. it takes help from the service proxy. The daemon of the service requester forwards the service request to the proxy and the service proxy intersects a service request, identifies the object(s) for the requested service, then fetch the functions from the objects on behalf of the requester and then finally serve it. A pictorial overview of the scenario is shown in Figure 3. The service proxy

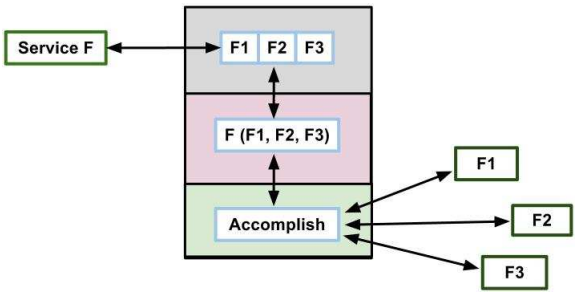

Figure 3: How service daemon works in a service proxy

maintains the functional information about all objects in the vicinity. It runs an algorithm to find objects which can serve a request or approximately serve the purpose. So a service proxy acts as a search engine (searching is also a service) with required cognitive abilities such that it finds an approximate service when the exact service is not available.

4.2.1 Search Convergence

When a device tries to find a service, it forms a neighborhood with some pre-specified metrics (for example network level hop count, physical distance etc.) Then it broadcasts the request in its neighborhood. If the device does not get reply from its neighbors, then the request is forwarded to a service proxy. The service proxy then provides the addresses of the service providers. If the service providers are not known to the proxy itself, it consults with other service proxies. It might happen that a service proxy might be available within the neighborhood of a device. In that case, the proxy immediately provides the results. A device learns about service proxy after the booting phase. It broadcasts query to locate service proxy along with its own service capability (as a provider). When this request reaches a proxy, the proxy learns about the device capability and also returns own address. Although a formal protocol has not yet been defined, we can assume that all message duplicates are discarded in the network.

4.3 Learning and Situation Awareness

Learning capability and situation awareness is one of the important factor that helps distributed search process for approximate service provisioning. From the system perspective, situation means any state within the time and space volume of the system [17]. The service daemon need to be aware of the situation from the perspective of service request handling. During the search process for service provider, a neighborhood is formed to broadcast search query. Based on the received responses, sometimes an approximate service provider needs to be chosen. These decision-making consider the current situation. For example, when a document needs to be printed, a printer is searched within a building/house or within a locality or within a city area depending on the urgency of the printing. Suppose, a color graph need to be printed by an employee in a office to show his boss within 5 minutes. In that case, a printer located within the same building can be used even if it is not a color printer. On the other hand, if he can show the graph on the next day, he can visit a printing shop for a proper color print.

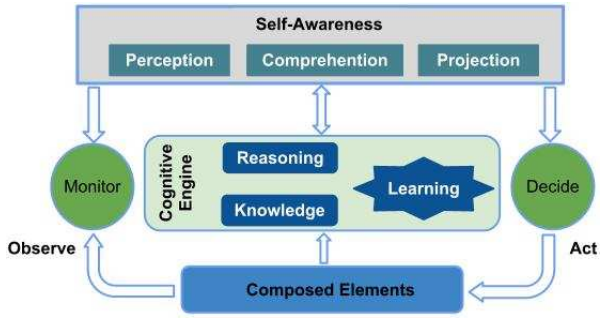

Figure 4: Situation awareness and Learning mechanism at a service proxy

The process of situation awareness is shown in Figure 4. A framework such as in [10] can also be used in conjunction to the Self-awareness module for better situation assessment. In order to percept the situation the cognitive engine of the service daemon should detect/monitor relevant elements/parameters. Once the new state (situational parameters) is detected then the service daemon should understand the significant of this changes. Once the action is taken this new situation and corresponding action should be projected into its learning mechanism.

4.4 Service Accomplishment Model

The Service lookup model (Figure 5) is closely coupled with the service daemon and works as following:

4.4.1 Specification of a Service

When a user or an application needs a service, the requirements for the service must be specified. The request contains context of the service, explicit requirement, etc. The device

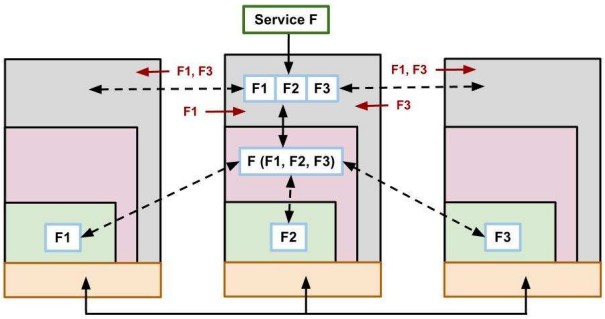

Figure 5: Service Identification with the help of Service Daemon

itself may ask for further information (if required) about the desired service while serving. Service requests are handled at the search level of the service daemon. If the daemon explicitly knows the object which can provide the service, then the request can be sent directly to it. But there can be other situations such as - (i) an exact (in its desired form) service provider is not available, but some alternatives are available, (ii) an exact service is available but a single object is not capable of serving the complete request, and (iii) multiple alternatives are available for a service (all of them may not be exact).

4.4.2 *Translating and Splitting a Service request into Functions*

In these complicated scenarios (mentioned in the previous section), to take a proper action on the service requirements, a highly cognitive mechanism must be in place which some devices might lack. On the other hand, the service proxy breaks a request into multiple smaller functions to find a suitable service provider for each functionality.

A service can be composed of a multiple (individual) functionality. In that case, the service need to be broken down into smaller functions. Let $f_1, f_2, ..., f_N$ are functions where each function is from one or more devices. In composing an approximate service, however, some functions may be "approximated" and some may even be missing altogether. This mapping of exact functions to approximated functions is shown in [19]. The missing functions are compensated by using alternatives found in the surroundings. Further, the alternatives may not be exact and thus, the capabilities available with the objects in the surroundings have to be compared with the requirements.

4.4.3 *Object Selection and Mapping with functions*

As said before, a service can be a conglomeration of functions, then each desired functions need to be mapped with an object which can perform the task. The mapping of tasks and objects is done at the compose level.

4.4.4 *Function Scheduling*

After deciding about the objects that can perform the required functions, the scheduling of each function need to be done because output of a function might be required by another function. Scheduling is important in order to maintain the dependencies among the sub-tasks/functions. Scheduling is done at the compose level as well.

4.4.5 *Communication with actual provider(s)*

After mapping the desired functions and scheduling them for different service providers, it is time to get the results. So the selected objects are communicated according to the schedule. The accomplish level of the service daemon takes care of the connection heterogeneity, i.e. objects with different communication capabilities. The communication interfaces for each object is known to service proxy (via learning). If the interfaces are not known, it is learned during runtime.

4.4.6 *Accumulating replies*

Each object is supplied with a set of inputs and the outcome is sent to the service daemon. Initiating communication (via known interfaces) and collecting output of each individual functions are done at the accomplish level.

4.4.7 *Assembling the functional outputs into a single service*

As discussed earlier, a service can be a conglomeration of multiple functions. Thus the output of each functions need to be assembled to get the final service.

4.4.8 *Serving the request*

Finally, the requested service is delivered to the user or to the application. The service proxy hides all details about breaking/assembling of functions from the user/application. The user/application simply requests for service and gets the service.

4.5 Feasibility of using Service Proxy

A centralized service proxy can be overloaded with the information about virtual objects. Then searching its database would also become a heavier task. Another concern is whether anyone can search an object or a service. This will become a major concern from the aspect of security and privacy. These challenges can be addressed by using a smart service proxy scheme. We envisage that there will not be a single service proxy. A layered structure of service proxies can be deployed. A lower tier service proxy will store information about the virtual objects in its vicinity only. Moreover it will store a small amount of information. As finding a service does not require a milli-second granularity (in most of the cases), the service proxy can ask an object for more information about it online. When the service proxy is not able to locate an object, it learns about other (higher level) service proxies and forwards the queries to them. When a user sends a request, his signature is attached with the request. This will verify an authorized request. If required, the service-proxy can ask further authentication information.

4.6 Advantages of Approximation-Service approach

An approximate service might not always meet the user requirements. Moreover, it can sometimes trigger a false positive. Thus this approach may not be beneficial. However, there is a set of advantages which can be achieved by using approximate services. First of all an approximation will come into place only when the exact service is not available. It will never replace an exact service. Other advantages are: (1) better availability of services to the users; (2) resources are used in a more efficient way; and (3) a large number of users can be served with varying needs.

5. CONCLUSION AND FUTURE WORK

In this paper, we have introduced Approximate Service and a simple distributed model for service lookup. We are motivated towards the use of such a new paradigm. We have provided a real life example. We provided a series of blocks that are necessary to offer such a service. We identified various steps towards finding and providing an approximate service. We plan to extend this work with a complete framework and also provide nuances of such a framework. We believe that Internet of Things would enable and require as well new paradigm shift towards completely cognitive service provisioning.

6. ACKNOWLEDGMENT

We thank iCore project. This article describes work undertaken in the context of the iCore project, *Internet Connected Objects for Reconfigurable Ecosystems* (http://www.iot-icore.eu/). iCore is an EU Integrated Project funded within the European 7th Framework Programme, contract number: 287708. The contents of this publication are the sole responsibility of iCore project and can in no way be taken to reflect the views of the European Union.

7. REFERENCES

[1] BUTLER. http://www.iot-butler.eu. [Online].

[2] FIND. http://www.nets-find.net/. [Online].

[3] FIRE. http://www.ict-fire.eu/ and http://cordis.europa.eu/fp7/ict/future-networks/. [Online].

[4] iCore. http://www.iot-icore.eu. [Online].

[5] IoT-A. http://www.iot-a.eu/. [Online].

[6] IoT.est. http://ict-iotest.eu/. [Online].

[7] Nsf workshop on pervasive computing at scale (pecs). http://sensorlab.cs.dartmouth.edu/NSFPervasiveComputingAtScale/whitepaper.html. [Online].

[8] Soa reference architecture. Technical report, OpenGroup, Dec 2011.

[9] G. Baldini, R. V. Prasad, A. R. Biswas, K. Moessner, M. Etelapera, J. P. Soininen, N. Septimiu-Cosmin, V. Stavroulaki, P. Vlacheas, A. Georgakopoulos, and P. Demestichas. A Cognitive Framework for Realizing and Exploiting the Internet of Things Concept. In *27th WWRF Meeting, Dusseldorf*, October 2011.

[10] Y. Fischer, A. Bauer, and J. Beyerer. A conceptual framework for automatic situation assessment. In *Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2011 IEEE First International Multi-Disciplinary Conference on*, pages 234 –239, feb. 2011.

[11] J. Hoebeke, G. Holderbeke, I. Moerman, M. Jacobsson, V. Prasad, N. I. C. Wangi, I. Niemegeers, and S. H. de Groot. Personal Network Federations. In *15th IST Mobile & Wireless Communications Summit, Myconos, Greece*, June 2006.

[12] M. Ibrohimovna and S. H. d. Groot. Policy-based hybrid approach to service provisioning in federations of personal networks. In *Proceedings of the 2009 Third International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, UBICOMM '09, pages 311–317, Washington, DC, USA, 2009. IEEE Computer Society.

[13] M. Jacobsson, I. Niemegeers, and S. Heemstra de Groot. *Personal Networks*. John Wiley & Sons, Ltd, 2010.

[14] N. Jefferies. Global vision for a wireless world. In *Wireless World Research Forum, Helsinki, Finland.*, volume 18th WWRF meeting, June 2007.

[15] B. Jiang and H. Olesen. *Agent-based Personal Network (PN) service architecture*, volume 2, CD-ROM, pages 275–279. 2004.

[16] S. M. K. Lichuan Liu and M. C. Zhou. Virtual Sensing Techniques and Their Applications. In *Proceedings of the 2009 IEEE International Conference on Networking, Sensing and Control, Okayama, Japan,*, March 26-29 2009.

[17] N. Nwiabu, I. Allison, P. Holt, P. Lowit, and B. Oyeneyin. Situation awareness in context-aware case-based decision support. In *Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2011 IEEE First International Multi-Disciplinary Conference on*, pages 9 –16, feb. 2011.

[18] R Venkatesha Prasad, Chayan Sarkar, Vijay S Rao, A Rahim Biswas and I. Niemegeers. Opportunistic Service Provisioning in the Future Internet using Cognitive Service Approximation. In *28th WWRF Meeting, Athens, Greece*, April 2012.

[19] R Venkatesha Prasad, Ertan Onur, Vijay S Rao, Yunus Durmus, A Rahim Biswas and I. Niemegeers. Approximate Service Provisioning in an Invisible Network of the Future. In *27th WWRF Meeting, Dusseldorf*, October 2011.

[20] C. Thornton. General Principles of Indirect Sensing. http://www.sussex.ac.uk/Users/christ/papers/principles-of-indirect-sensing.pdf, August 2007. [Online].