The MediaSense Framework: Ranking Sensors in a Distributed Architecture

Jamie Walters, Patrik Österberg, Theo Kanter
Deparment of Information Technology and Media
Mid Sweden University
Sundsvall, Sweden
{*jamie.walters, patrik.osterberg, theo.kanter*}*@miun.se*

***Abstract*—The evolution of mobile applications and services, largely enhanced by their ability to respond to changes in a user's situation is key driver towards an *Internet of Things*. With a size expected to exceed the current Internet, solutions are required for self organisation based on relevance and importance. This, underpinning new applications and services exposed to reliable and evolving context information around a presentity. One such element is the ability to rank heterogeneous and distributed sensors in response to real time interaction between users and the digital ecosystem. In this paper, we consider one approach to calculating sensor ranking based on their general usage patterns. We present it relevant to our work in progress MediaSense framework, showing the ability for it to be implemented without the need for any centralized coordination.**

***Keywords*-Sensor Ranking; MediaSense; Context-Awareness; Presence**

I. Introduction

As with any typical day within an urban environment, people are constantly on the move for business or pleasure. Within such a future cityscape, there exists a digital ecosystem capable of providing enough information in order to derive support for services wishing to affect changes or deliver experiences to a user based on some context. This includes audio-visual devices, internet connections, or a range of sensors such temperature, humidity or even traffic and air quality.

William, a 10 year old child with a mobile phone would be able to connect to, and derive representations of, context from these points in order to support his applications. His mother has a application on her mobile phone which reports on William's current context situation; i.e., is it too warm, too cold, raining, etc. He should therefore be able to connect to the most accurate and reliable sensors reporting the current outdoor temperature in order for his mum whether he is safe or not.

Current solutions implementing context awareness services rely on the availability of fixed information points from which to derive indications of context. William could be connected to the SenSei architecture [1], which enables the deployment of applications and services in response to his context. He would, however be given the nearest sensor, or the sensor connected to his infrastructure in order to derive a temperature value. While the addition of new sensors would be made available to William, they would not be recommended based on any metric outside of being a part of his domain or infrastructure. If he is in a room, he would like to be assigned the temperature sensor attached to the room.

In a wide and heterogeneous *Internet of Things* [2], there can and will exist multiple sensors in the same locality being offered as sources of context information. William must be able to derive enough information in order to, whether manually or autonomously, select the most suitable sensor for reporting his temperature.

One such element of information is the current ranking, the implied reputation the sensors available to William. An indication of reputation could be derived from the behaviour of the users; continually connecting to and using the sensor as a source of context information. An accurate and reputable sensor, by any measure, would more likely be chosen and used over a sensor that is considered to be inaccurate and usually unavailable. For William, the temperature sensor attached to the building might not be the best sensor, as it might situated close to the heating radiator and reporting a reading that is several degrees higher than the actual temperature in the room. However a sensor temporarily placed in the room would not be made available by [1] which would first require it to be added to profiles and made available, which is not be feasible in dynamic environments. Therefore, there exists, a need to be able to identify changing patterns in user behaviour, such as most users not connecting to the existing sensor but to the alternative, i.e., a sensor ranking approach complementing the sensor proximity approaches mandated by [3].

Current ranking approaches such as Internet search engines consider the theory of connected things, however relative to static document content on the Internet. A document's connectivity determines its relevance. This concept of ranking has been explored and used both in a centralized solutions [4] as well as distributed solutions [5]. However, centralized solutions such as Google index only a tiny portion, less than 10 billion of the estimated 550 billion pages, on the relatively static Internet [6], [5]. Any attempt to apply such a centralized solution to the ranking of sensors in an *Internet of Things* would be undermined by their ability to scale well. Distributed solutions such as [5] which are

 ISBN: 978-1-61208-127-4

Figure 1. The MediaSense Framework

based on the PageRank [7] concept would not scale well to accommodate highly dynamic document sets. Current *real-time* searches are realized by targeting known content providers, an approach that could not scale to accommodate the vast and mostly ad-hoc nature of a connect things infrastructure.

Latencies with respect to scalability could not guarantee William's mother freshness or accuracy with regards to the information being used in her judgement. The MediaSense project in its ongoing work, realises the need to create solutions capable of a real-time distributed ranking algorithm.

II. The MediaSense Framework

The MediaSense framework seeks to create an enabling platform for the *Internet of Things*. At its center are presentities [8] and their interaction with the *things* within a digital ecosystem. Supported by an overlay, this solution enables the distributed provisioning of context information reflecting such real world interactions. Figure 1 illustrates the composite layers; each contributing to this realisation. Enabling an interactive environment with respects to a user's state, interactions and preferences. The framework is comprised of the following layers:

1) Overlay: The MediaSense Framework is underpinned by a distributed overlay network used to maintain backbone communication as well as providing an indexing mechanism for information that must be persisted among nodes. As with typical peer-to-peer protocol implementations, the nodes participating within the overlay act as entry points for application and services wishing to execute a query over all or a subset of the participating nodes. Early implementations employed a Chord [9] based overlay. This had the inherent drawbacks of distributed hash table (DHT) based implementations with respect to supporting range queries. Citing this and the advantages of P-Grid [10] over DHTs, we have migrated the overlay to a P-Grid based implementation.

2) DCXP: Residing immediately on top of the overlay is the Distributed Content Exchange Protocol (DCXP). As

Table I
The Primitives of the Distributed eXchange Protocol

REGISTER_UCI	Registers a UCI with the overlay making it available for use.
RESOLVE_UCI	Resolves a UCI to the node which is responsible for it.
GET	Retrieves the current context value from the node responsible for a UCI. The reply is sent using a *NOTIFY*.
SUBSCRIBE	Submits a subscription request to the node responsible for a UCI. The node in turn sends a *NOTIFY* message containing the current context value, either at regular intervals or when the value changes.
NOTIFY	Dispatches the current context value associated with a specified UCI to an interested node..
TRANSFER	Reassigns the management responsibility of a resource to another node. This might be full or partial responsibility, where the requester recreates a local copy of the resource permitting improved real time performance.

with the early implementation of our architecture, this layer implements the core protocols employed in the provisioning of context information. However one key departure is that the protocol is no longer utilized in the maintenance of the overlay; this being completely managed by the P-Grid overlay itself. As a result of this, network composition and state is abstracted from the protocol layer. With this key modification, the DCXP primitives have been adjusted, with the new set of primitives listed in Table I.

We introduced a single new primitive, the *TRANSFER* primitive. This provides the ability to relocate context resources to nodes closer to where their demand is greater. This is in an effort to reduce network messaging overhead, and the considerable demands that can be placed on nodes responsible for a context resource. Such an action could be achieved autonomously for load balancing or in response to application requirements.

3) Persistence: Persistence is offered at each node in the form of object-oriented databases. The purpose of this is two fold: firstly, it provides a persistence mechanism for context information generated by sensors local to the node. These values are made available in response to DCXP *GET* and *SUBSCRIBE* requests, providing a source of sensor information to the nodes in the overlay. Secondly, it provides a persistence mechanism for the objects created in the *Object Layer* and enable the searching and browsing of context objects over the framework. This enables a single view across all the entire collection of information in the overlay, similarly to a distributed database.

4) Object Layer: The object layer exposes all the underlying information as a collection of objects that are accessible and can be used to realise the provisioning of an application or service in response to context information. Objects as made available through the API which permits

 ISBN: 978-1-61208-127-4

the definition of the *object-predicate-object* relationships constituting the Context Information Integration Model (CII) [11].

Objects are accessible across the entire framework by virtue of the underpinning overlay and its ability to *TRANSFER* resources between end-points. Objects may be comprised into schemata relative to presentities, permitting an easy reference to the resources relative to a presentity and thus available to an application or service.

5) Schema: We introduced the concept of a *Context Schema* [12], defined as:

The collection of information points associated with and contributing to a presentity's current context

where an *Information Point* is defined as:

Any source providing information about the context of an entity or any sink capable of accepting an input effecting changes to an entity's context

Within the schema layer, such a schema is attached to a presentity and encapsulates all the information points and the relationships related to a presentity. An application or service with a requirement to deliver some user-context centric experience subscribes to the current schema description; it realises a collection of information objects underpinned by a *publish/subscribe* interface to the end points described by the schema. As a presentity traverses a connected things infrastructure it discovers new entities and consequently updates its schema to reflect this. As a result, all subscribing end points receive an updated schema and can adjust their services to accommodate this. For simplicity, we refer to information points as *sensors* within the remainder of this paper.

6) API: The API layer presents itself as a facade to be utilized by application developers in accessing the framework's functionalities. It masks the complexity of lower layers and their interactions, enabling users to focus on developing context objects, applications or services; having them transparently shared across the network with relative ease.

III. Ranking Sensors in an Internet of Things

One problem being addressed by the MediaSense project is the ranking of sensors within a heterogeneous and distributed landscape. While traversing an *Internet of Things*, users will be excepted to encounter masses of information sources such as sensors. Such sensors, whether physical or virtual are positioned to the user as points from which to derive dimensions of context which can be used to support a multitude of available applications and services. A supporting solution should therefore should be able to provide application developers and users with as much information as possible in order to select the most relevant and recommended sources available. We therefore look at

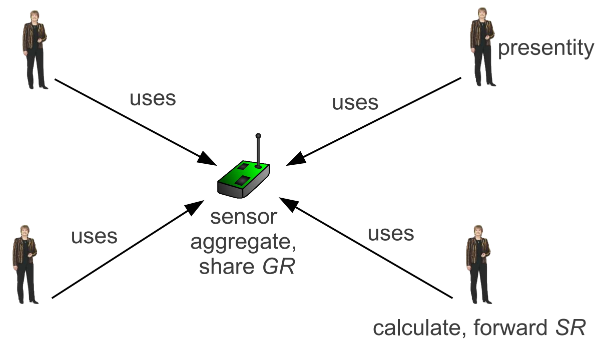

Figure 2. Calculating Sensor Ranking

approaches to adding useful metrics to sensors enabling applications and service providers to rank and select suitable sensors from any groups of sensors encountered.

A. Approach

The schema objects described in Section II-5 permit presentities to create collection of sensors, contributing to and expressing the context over a presentity. Using this approach, we are locally aware of all the instances where a sensor s has been utilized by a presentity P in schema construction. We also know all the schemata that have been used by this presentity P and can therefore derive some representation of the importance of s relative to P. We argue that such a value,represents the localized ranking of s and as such the node where s resides is therefore able to collect and aggregate these values, indicating an overall, near global, ranking value of s.

B. Localized Ranking

Our ranking algorithm consists of two main components, illustrated in Figure 2. Firstly we need to determine the local ranking value for s with respects to P_i. We then need to aggregate the global ranking value for s. In approaching this problem, we adapt a modified version of the *Inverse Document Frequency* algorithm [13]. This algorithm, initially used to calculate the importance of a query term with respect to a document corpus, provided a simple but representative metric for ranking documents with respects to a search query.

The algorithm shown in Equation (1) is modified with respects to a sensor s, a schema r and a presentity P. A sensor in a schema is considered to be analogous to the query term in a document and is expressed as follows:

$$SR_{si}{}^{(P)} = log \frac{|R|}{\{r : s_i \in r\}} \tag{1}$$

Where SR is the sensor ranking of the sensor s, R is the corpus, the total collection of schemata relative to presentity P with r being all the schemata relevant to P containing a reference to sensor s. This provides us with a representative metric as to the importance of sensor s relative to P. We consider further, that there exists scenarios where some presentities will be less dynamic or mobile with respect to s. Such an example might be a sensor located in a store; the employees working in the store will by default almost always utilize the sensors that are local to the store accounting for a disproportionately higher value for:

$$\frac{|R|}{\{r : s_i \in r\}} \tag{2}$$

In such scenarios, all stores within a shopping area would have high values, granted solely by the employees themselves. Therefore, by taking:

$$log \frac{|R|}{\{r : s_i \in r\}} \tag{3}$$

we consider more dynamic presentities traversing an *Internet of Things*. A person that travels around the city interacts with more sensors subsequently creates more context schemas in fulfilment of service delivery. This is represented by larger ratio of R to $\{r : s_i \in r\}$. Such presentities we argue, indicate a more accurate representation of the ranking that should be associated with s, relative to the wider sensor ecosystem.

Another scenario being that we penalize malicious nodes may that might attempt to collude or independently attempt to inflate their rankings by creating a disproportionate number of schemas. In such scenarios, the result of Equation 2 would move closer to a value of 1. By taking its log, instead, we adjust this such that for $|R| = \{r : s_i \in r\} = 1$, $SR = 0$. Thereby having a very small effect on the sensor ranking.

C. Time Limited Localization

While Equation (1) permits the calculation of ranking over the entire interactions of P, the need will arise to be able to calculate sensor rankings at some given point t in time. Such a scenario would be useful when trying to rank sensors that are in use at an event or at a situation occurring in a localized area. Here, we could calculate ranking limited by some time duration of interest, t.

$$SR_{si}{}^{(P)} = log \frac{|R_t|}{\{r : s_i \in r_t\}} \tag{4}$$

Each presentity now has a value for its ranking of s, both historically or relative to some interesting duration. This metric permits the presentity to evaluate the usefulness of s. This being available should it again encounter s either isolated or as a potential sensor amongst several other sensors.

Algorithm 1 Ranking Sensors and Information Points

```
loop
  {at the local node, i}

  determine the size of the local corpus R_i of schemata
  for all information points s attached to P_i do
    determine the ranking value with respect to R_i
    assign this as the local ranking value SR_i
    forward this to the global domain owner D
  end for
  {at each domain owner, j}

  for all information points s_i residing at D do
    aggregate the all values received for SR_{s_k} to find GR_{s_k}
    calculate the new value for the domain rank DR_j from all GR_s
  end for
end loop
```

D. Global Aggregation

The second component of our approach is a global aggregation of all the local ranking values SR assigned to s. To achieve this, we calculate the Global Ranking GR by finding the sum of all SR of s such that:

$$GR_s = \sum_{k=0}^{n} SR_k^{(P)} \tag{5}$$

This value is continually calculated as new schemas referencing s are created. We however take into consideration the owner of s, the presentity or domain where it resides or to which it belongs. This we regard as the domain D and assign it a value equal to the average ranking of all the sensors belonging to D. This we call DR, the ranking of D. This value is important to us as it would permit us to identify more connected and important spaces such as domains, buildings or just a collection of deployed sensors. We calculate DR as:

$$DR = \frac{\sum_{k=0}^{n} SR_k^{(P)}}{k} \tag{6}$$

The resulting values derived above, can be used as indicators or relevancy or importance of sensors in an *Internet of Things*.

IV. Approach on a Distributed Architecture

Within a distributed architecture, the implementation of such a sensor ranking approach gains the best implementation with respect to performance and its ability to scale. A presentity would likely not reside on the same node as the sensor that it is trying to use.

Our distributed solution, could be implemented on the MediaSense Framework making use of the schemas and the underlying overlay for messaging support. A node using a sensor in a schema in order to represent the context information for a presentity, calculates the localized ranking value and forwards it to the node responsible for the sensor. The node owning the sensor, aggregates the sensor ranking and makes this available on the overlay to any other interested nodes. This is summarised in Algorithm 1. The benefit of this approach is that sensors calculate values locally with no centralization needed, deriving its scalability properties from the underlying infrastructure.

V. Conclusion and Future Work

In an *Internet of Things*, users will require useful metrics in order to understand the digital ecosystem in which they are embedded. Apart from the sensor information themselves, we require metrics that construct an overview of a sensor's ranking. We have shown above that we can adapt existing document ranking algorithms towards solving this problem, providing algorithms that can derive these metrics representative of a sensor's importance in a context centric infrastructure.

This is achievable through additional extension of MediaSense to include a distributed context model and the introduction of context schemas attached to presentities.

We further elaborated on how such a set algorithms can be implemented relative to the distributed approach of the MediaSense framework.

Future work on this area would involve the inclusion of sensor proximity factor with sensor ranking to improve the filtering of undesirable sensors. The reputation of the presentities awarding the ranking values could be taken into consideration as well as the reputation on the nodes calculating the global ranking value.

Acknowledgment

This research is partially funded by the regional EU target 2 funds, regional public sector, and industry. The authors would like to thank Victor Kardeby, Stefan Forsström, Enrico Saviolli for their feedback and contribution.

References

[1] M. Presser, P. Barnaghi, M. Eurich, and C. Villalonga, "The SENSEI project: integrating the physical world with the digital world of the network of the future," *Communications Magazine, IEEE*, vol. 47, no. 4, pp. 1–4, Apr. 2009. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4907403

[2] H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelfflé, "Vision and Challenges for Realising the Internet of Things," *Cluster of European Research Projects on the Internet of Things (CERP-IoT)*, no. March, 2010. [Online]. Available: http://docbox.etsi.org/tispan/Open/IoT/20100601 IoT Conference/ALL SLIDES/EU_IOT_Clusterbook 2009.pdf

[3] J. Hong and J. Landay, "An infrastructure approach to context-aware computing," *Human-Computer Interaction*, vol. 16, no. 2, pp. 287–303, Dec. 2001. [Online]. Available: http://portal.acm.org/citation.cfm?id=1463119

[4] "Google http://www.google.com," p. http://www.google.com, 2010. [Online]. Available: http://www.google.com/

[5] Y. Zhu, S. Ye, and X. Li, "Distributed PageRank computation based on iterative aggregation-disaggregation methods," in *Proceedings of the 14th ACM international conference on Information and knowledge management*. New York, New York, USA: ACM, 2005, pp. 578–585. [Online]. Available: http://portal.acm.org/citation.cfm?id=1099554.1099705

[6] J. Li, B. Loo, J. Hellerstein, M. Kaashoek, D. Karger, and R. Morris, "On the feasibility of peer-to-peer web indexing and search," *Peer-to-Peer Systems II*, pp. 207–215, 2003.

[7] K. Sankaralingam, S. Sethumadhavan, and J. Browne, "Distributed pagerank for p2p systems," in *High Performance Distributed Computing, 2003. Proceedings. 12th IEEE International Symposium on*. IEEE, 2003, pp. 58–68. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1210016

[8] H. Christein, *Distributed Communities on the Web*, ser. Lecture Notes in Computer Science, J. Plaice, P. G. Kropf, P. Schulthess, and J. Slonim, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, Dec. 2002, vol. 2468. [Online]. Available: http://www.springerlink.com/index/10.1007/3-540-36261-4

[9] I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, vol. 31, no. 4. ACM, 2001, pp. 149–160. [Online]. Available: http://portal.acm.org/citation.cfm?id=964723.383071

[10] K. Aberer, P. Cudré-Mauroux, A. Datta, Z. Despotovic, M. Hauswirth, M. Punceva, and R. Schmidt, "P-Grid," *ACM SIGMOD Record*, vol. 32, no. 3, p. 29, Sep. 2003. [Online]. Available: http://portal.acm.org/citation.cfm?doid=945721.945729

[11] F. Dobslaw, A. Larsson, T. Kanter, and J. Walters, "An Object-Oriented Model in Support of Context-Aware Mobile Applications," *Mobile Wireless Middleware, Operating Systems, and Applications*, pp. 205–220, 2010. [Online]. Available: http://www.springerlink.com/index/UH0745GR616N7387.pdf

[12] J. Walters, T. Kanter, and R. Norling, "Distributed Context Models in Support of Ubiquitous Mobile Awareness Services," in *The 2nd International ICST Conference on Sensor Systems and Software*. Miami: Springerlink, 2010.

[13] S. Robertson, "Understanding inverse document frequency: on theoretical arguments for IDF," *Journal of Documentation*, vol. 60, no. 5, pp. 503–520, 2004.