Vehicular Network Enabling Large-Scale and Real-Time Immersive Participation

Theo Kanter, Rahim Rahmani, Yuhong Li, and Bin Xiao

Department of Computer and Systems Sciences
Stockholm University
{Rahim,kanter,yuhongli,xbin}@dsv.su.se

Abstract. This paper presents a system and mechanisms enabling real-time awareness and interaction among vehicles connected via heterogeneous mobile networks. Information obtained by vehicles is considered as the centre in our system. Vehicles are organized dynamically in overlaid clusters. In each cluster, vehicle-related information is pushed in time. As a network node, each vehicle has the function of content abstraction and distribution. Through processing and abstracting the sensed data, various vehicle-related information are organized and denoted in hierarchical names at each node. The data are transmitted and forwarded using protocols accordant with the characteristics of the content. In this way, large-scale and real-time information exchanges among vehicles are realized. Part of our system has been implemented and tested. An open source platform providing standard sensor and actuator API can be provided.

Keywords: Internet of Vehicle, Information-centric networking, MediaSense.

1 Introduction

Internet of Vehicles (IoV) has been acknowledged as an important part of people' daily life. It has been identified as a key technology for increasing road safety and transport efficiency, and providing infotainment in the wireless and mobile environment. By connecting people with vehicles, vehicles with vehicles, and vehicles with environments, various information and services can be obtained.

At a high level, the goal of the IoV is to enable the efficient exchange of information among the vehicles, which can act as both information producers and consumers. However, due to the highly dynamic topology and non-uniform distribution of vehicles as well as the large amount of information in IoV, exchanging diverse information efficiently among the vehicles is a great challenge.

First, with the development of IoV and the increasing number of vehicles, more information needs to be sensed and transmitted. For example, for safe driving on the road, various types of information are needed, as shown in Figure 1. The safety data and other road conditions need to be exchanged among the vehicles in the similar situation. The vast number of vehicles may exacerbate the amount of information by orders of magnitude. All the information data need to be processed, organized and

R.C.-H. Hsu and W. Shangguang (Eds.): IOV 2014, LNCS 8662, pp. 66–75, 2014.

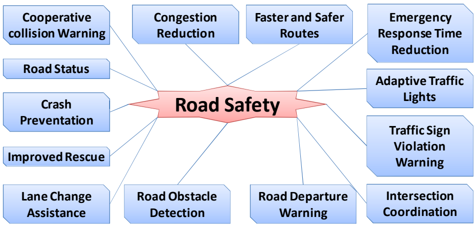


Fig. 1. Information for road safety

abstracted in each vehicle in order to be cached and transmitted among the vehicles efficiently and economically.

Secondly, the information related to vehicles has time and spatial constraint. The data transmission mechanism should consider the characteristics of the information.

Thirdly, real-time awareness is more important in IoV. Local experiences and interaction include sources and sinks globally. All of which may require instant responses. In addition, infotainment becomes more and more important for people, for example, for those having long journey on buses. In this case, on-line games with friends at home maybe interesting for them. Thus, large amount of video and audio data needs to be transmitted in time.

Fourthly, the IoV has highly dynamic topology and non-uniform distribution of nodes. Therefore, a scalable data dissemination mechanism, which can adapt to frequent topology changes with variable vehicle density, and can satisfy certain real time requirement is necessary.

Current research on IoV has concentrated on routing, service performance, security and mobility management in heterogeneous wireless networking environment etc. [1]. Various results have been obtained regarding establishing physical communication links among the vehicles using ad-hoc technique [2][3]. Compared with these work, we concentrate on vehicle-related information distribution. Work [4] [5] has introduced the idea of Information-centric Networking (ICN) to disseminate data in vehicular networks. However, how to organize and store the large amount of data sensed by vehicles efficiently and disseminate the data in real-time and through the large-scale vehicular network has not been discussed in detail.

In this paper, we concentrate on how to provide a real-time IoV from the point of view of information distribution in large-scale. We propose to use ICN and ontology techniques to disseminate efficiently information in the context of IoV. A new method for constructing the IoV and a new architecture of end-device are proposed. Using this architecture, the sensed data by each vehicle are abstracted and organized in named contents. Through the publication and subscription mechanisms, vehicles can obtain their interested content in less delay. The rest of the text is organized as follows. First, the vehicle network system enabling large-scale immersive participation is introduced and the architecture of end-devices is presented in section 2.Following this in section 3, mechanisms for information abstraction and organization, as well exchanges among vehicles are also addressed. Then the implementation and tests of the system are depicted in section 4, and conclusions are made in section 5.

2 Vehicular Network System Enabling Large-Scale Immersive Participation

2.1 System Architecture

The whole IoV system is consist of logical clusters, as shown in Figure 2. A logical cluster implies that vehicles might reside remotely physically but clustered logically based on the same or similar contexts. This allows resources (data, services) to be shared among different physically distributed vehicles more easily. Each cluster is identified by a context.

A cluster is constructed when a vehicle publishes a new context. Any vehicle can join a cluster dynamically depending on whether it is interested in the context. A vehicle can join different clusters simultaneously according to the contexts it is interested. Several contexts can be shared in one cluster.

Contexts are keys for forming the various clusters. In our system, a context is defined as a certain category of vehicle-related information. The contexts have life period, so clusters can dissolve dynamically.

Each cluster is maintained autonomically. According to the characteristics of a context, a context is published periodically by the vehicle that creates it. Vehicles interested in this context will record the context name in their own Subscribed Table. Afterwards, when the vehicle receives data with the same context, the data will be delivered to the corresponding consumer part of the vehicle. If a vehicle does not interest in the context anymore, it will delete the corresponding entry in the Subscribed Table. Then the data packets will be forwarded or discarded instead of delivering to the consumer part. The context entry in the Subscribed Table will also be deleted after it expiries.

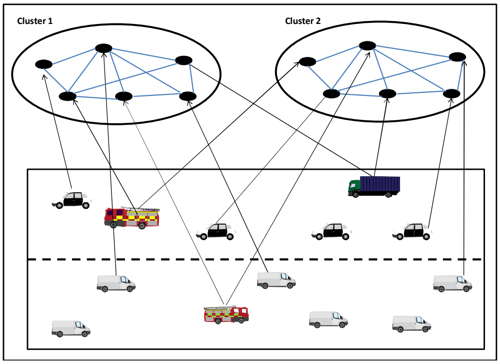


Fig. 2. System architecture

In order to be able to exchange information among the vehicles in real-time, contents are pushed in every cluster. In other words, no procedure of content requirement from a vehicle is needed. But physically, the data packets can be forwarded by any vehicles within the wireless radio connection, no matter if the vehicles belonging to the same cluster or not. Through this way, the logical clusters increase the scalability of the IoV and reduce the complexity for distributing the contents.

2.2 Functions of Vehicle Nodes

In order to achieve the above goal, the following design principles are followed:

- Flexible construction and adaption of the overlay clusters. Self-adaptation with little or no human intervention in the dynamic environment is needed. Moreover, the adaptation should be based on the contexts.
- Efficient management and distribution of contexts. Contexts should be well organized. The underlying network infrastructure should not be overloaded due to the exchanges of context data in time during the operation of the vehicles.

Hence, a modular architecture comprising four blocks is suggested for each vehicle device to allow for flexibility and extensibility while additionally supporting a clear separation of concern. Figure 3illustrates the architecture of a vehicle as a networking node.

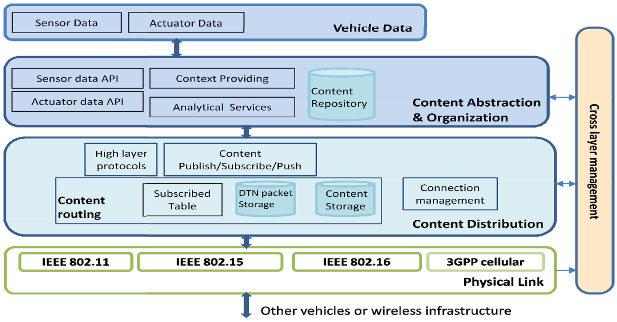


Fig. 3. Architecture of a vehicle node

The architecture consists of four blocks, namely Vehicle Data, Content Abstraction&Organization, Content Distribution and Physical Link.

Figure 4 illustrates the Vehicle Data block. It has mainly two functions. One is data sensing. Various data can be obtained through the event sensing and monitoring functions integrated in the vehicles, such as the current position and status of the vehicles, the geographical environment etc. The other is to receive information from other vehicles. The information can be used to configure the vehicle or provide for the people in the vehicle as infotainment. We call this part of data actuating. Both sensor data and actuator data contact with Content Abstraction & Organization block through the standard API provided by the Content Abstraction & Organization block.

Content Abstraction & Organization block is responsible for processing, abstracting and organizing the sensed data. It is the core of the system. In this block, MediaSense platform [6] provides access to all types of sensors and actuators via gateway, by translating communication protocols or methods used by sensors or actuators, or which is connected into the MediaSense through the API interface. The data is analyzed and abstracted by MediaSense using data mining and machine learning techniques. New information can also be created according to the contexts based on the ontology technique, such as traffic jam indication or danger warning, which can be passed to the Vehicle Data block to further conduct the vehicle. All the information are named by MediaSense, and stored in the content repository.

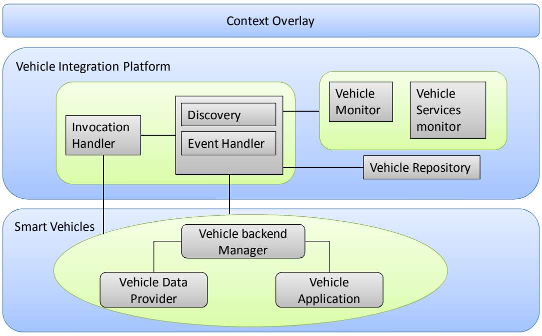


Fig. 4. Integration platform providing vehicle-related data

The Content Distribution block is responsible for disseminating the content data in the overlay clusters, using the most suitable application layer and networking layer protocols as well as physical wireless access technologies. The contents stored in each vehicle are disseminated in terms of the characteristics of the contents using ICN technique. According to the characteristics signed by the MediaSense, the content data can be encapsulated in different application layer protocols such as DTN, P2PSIP, XMPP, and forwarded using different networking layer protocols, such as immediate flooding or opportunistic forwarding (DTN mode).

A vehicle may have several physical interfaces supporting different wireless access technologies, such as WiFi, 3GPP cellular or Bluetooth. The link status of different physical interfaces, such as the signal strength etc. are collected by the cross-layer management module, which will be used by the Connection Management module in order to transfer data in the most suitable link.

3 Content Organization and Distribution

3.1 Content Organization and Abstraction

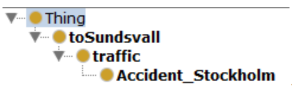
The information at each vehicle is organized hierarchically. At the top level, two types of contexts are defined, namely network condition context and content context. A network condition context is generated periodically, including the physical network interfaces available, a list of known connected nodes, a list of out-of-contact nodes, by overhearing network traffic and detecting the retransmission failure towards known destination. The network condition contexts are used for the efficient information distribution. Content contexts in our system mean the contexts for organizing a certain category of vehicle-related information. For example, /toSundsvall means the aggregation of all the information for vehicles driving to the city Sundsvall; whereas /toSundsvall/ traffic means the traffic context to Sundsvall.

A specific piece of information is named in a URL- like way hierarchically by using its context name plus the content name. For example, /toSundsvall/traffic/accident_Stockholm is the name of content that an accident happens near Stockholm in the direction to Sundsvall. We support variable context length (i.e., the number of components in a context name). As mentioned below, the longest name prefix matching algorithm is used for matching the context name for content distribution.

Besides the name, some characteristics of the information are also recorded together with the information, including the interval for disseminating the content periodically and the life time of the content etc.

The MediaSense platform is responsible for the context organization and abstraction. MediaSense abstracts data from all sensors into a generalized and standardized format. The ontology technique is used to create and organize the information according to the categories of the information, such as the position of the vehicles, the geographical environment, the status of the vehicles and the general traffic on the road etc. To support the description logic of ontology, OWL-DL [7], a branch of OWL, is used. OWL-DL is an ontology formatting language standardized by W3C [8], which provides rich syntax to annotate the raw content data with description logic. Vehicle-related data, such as the position and the geographical environment and so on are formed as ontology entities using OWL/XML files. Each of the entity is slotted by variables to present the current features within the raw context under certain problem domain.

For example, by utilizing the hierarchical mechanism of ontology, we carry out layered marking to the contents. Thing, as the content context, is the super class for the whole ontology. Taking "/toSundsvall/traffic/accident_Stockholmas" as an example, under the super class "toSundsvall" is defined as a subclass to mark the destination of a certain travel, as shown in Figure 5. The content file is marked accordingly using OWL/XML file. Each hierarchy is annotated as a class in the file system based on object-oriented concept, where ontology based reasoning is supported. The contents are maintained in the content repository.




```
<?xml version="1.0"?>

<!DOCTYPE rdf:RDF [
    <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>

<rdf:RDF xmlns="http://www.owl-ontologies.com/Ontology1398761667.owl#"
     xml:base="http://www.owl-ontologies.com/Ontology1398761667.owl"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
    <owl:Ontology rdf:about=""/>
    <owl:Class rdf:ID="Accident_Stockholm">
        <rdfs:subClassOf rdf:resource="#Traffic"/>
    </owl:Class>
    <owl:Class rdf:ID="toSundsvall"/>
    <owl:Class rdf:ID="Traffic">
        <rdfs:subClassOf rdf:resource="#toSundsvall"/>
    </owl:Class>
</rdf:RDF>
```

Fig. 5. The hierarchy for a piece of message sample

3.2 Content Distribution

A scalable data dissemination mechanism is necessary, which can adapt to frequent topology changes with variable vehicle density and can satisfy certain real time requirement, taking network resources into consideration. Therefore, we use information-centric networking (ICN) technique [9][10] to realize the data dissemination in our architecture. One the one hand, ICN insulates fundamentally the data transmission from dynamic topologies through naming the information and routing/forwarding them according to their identities. On the other hand, the in-networking caching feature of ICN can help to reduce the data response time and data traffic. The receiver-oriented operation solves the mobility issues internally, and the natural support for multicast make the data exchanges more fault tolerance. However, the time and spatial constraint of the information, as well as the highly dynamic topology and non-uniform distribution of nodes must be considered.

In addition, the receiver-oriented chunk based data transmission (i.e., pull-based) designed by ICN, may degrade the performance of IoV. For example, normally the information about an accident needs to be broadcasted to a certain geographical scope immediately. Therefore, we introduce also the push-based data dissemination mechanism in our system.

Hence, three data exchange procedures are used in our system.

- Publishing: when a new content context is created, the corresponding vehicle will broadcast the context. The name of the content together with its life time and publishing interval is included in the message. To be more efficient, the message is broadcasted periodically and storing and forwarding mode is used.
- Subscribing: when a vehicle is interested in a certain category of contents, it will register the context name in the Subscribed Table.
- Pushing: whenever a vehicle has information to be announced, it broadcasts the information in its cluster. According to the features of the context, the broadcast can be done periodically in a certain time.

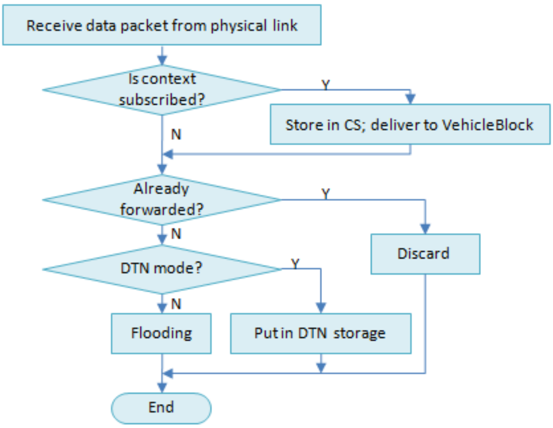


Fig. 6. Routing in content distribution

Figure 6 illustrates how data packets are routed by a physical vehicular node. When a vehicular node receives a data packet from the physical link, according to the content name in the data packet and the subscribed content context, it can decide if the data is that it wants. If yes, the data is delivered to the Vehicle Data block and cached in the Content Storage (CS) in terms of the context name. Otherwise, the node will check if there is any connection available and if the information should be forwarded in DTN mode according to the features of the context, and then flooding the packet through the physical link in terms of the results.

When a vehicle wants to send any data, the Content Publish/Subscribe/Push module will check the characteristics of the content and then encapsulate the data in a certain application and transport layer protocols, select a suitable physical link according to the network condition contexts and send out. For the moment, we support DXCP and DTN at the application layer.

To speed up the data forwarding, the Subscribed context Table (ST) is used, which records all the contexts it has subscribed. When the content name in the data packet matches an entry in the context table, the data packet will be delivered to the Vehicle Data block. Otherwise, it will be forwarded according to the network conditions. The algorithm based on hash table [11] is used in the system to implement the longest name prefix matching.

For the moment, we simply use flooding mechanism to ensure the reliable hop-by-hop data transmission for an ad-hoc ICN.

4 System Implementation and Tests

Part of the system has already been implemented and tested.

The core module–MediaSense [6][12] in the Content Abstraction & Organization block has already been implemented and launched as an open source IoT platform (www.mediasense.se) for scalable distributed context sharing and control. The Sensor and Actuator Data API has been tested by different IoT scenarios, such as energy profiling (for energy awareness), health monitoring (for medical status and alerts) and intelligent home automation. Currently the data sharing in MediaSense is implemented through the application layer overlay (P2P) techniques. We are now implementing the ICN mechanisms in order to disseminate the information more efficiently.

To validate the Content Abstraction & Organization block, we define an ontology as shown in Figure 7. Each white ellipse is a sub context domain, while each arrow shows the entity properties for each entity. Entity property is the way to mark the relations among entities, where each entity can be regarded as a property for its peers. The blue ellipse describes the constructions for the message in each context domain. According to a case scenario abstracted from the system, Position, Vehicle Status, Geo-information, Traffic-information and Infotainment are extracted out as the context domains, where each message can be marked with category under the construction format. A proof-of-concept prototype is written with Jess [13], which helps to establish the entity properties. AOWL/XML file is generated by a short Jess code, as illustrated in Figure 8.

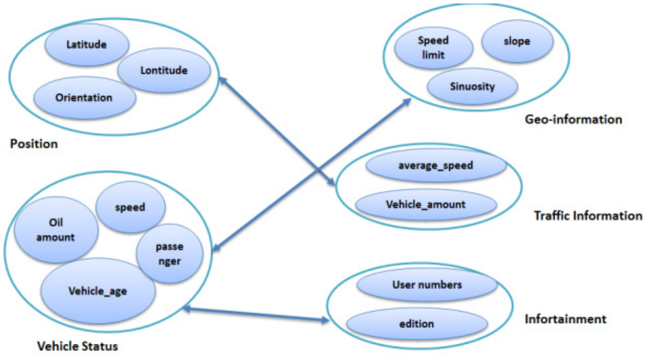


Fig. 7. An example of the created content and ontology

```
<?xml version="1.0"?>

<!DOCTYPE rdf:RDF [
    <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>

<rdf:RDF xmlns="http://www.owl-ontologies.com/Ontology1398340415.owl#"
     xml:base="http://www.owl-ontologies.com/Ontology1398340415.owl"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
    <owl:Ontology rdf:about=""/>
    <owl:DatatypeProperty rdf:ID="active">
        <rdf:type rdf:resource="&owl;FunctionalProperty"/>
        <rdfs:domain rdf:resource="#infotainment"/>
        <rdfs:range rdf:resource="&xsd;boolean"/>
    </owl:DatatypeProperty>
    <owl:DatatypeProperty rdf:ID="average_speed">
        <rdf:type rdf:resource="&owl;FunctionalProperty"/>
        <rdfs:domain rdf:resource="#traffic_inf"/>
        <rdfs:range rdf:resource="&xsd;int"/>
    </owl:DatatypeProperty>
    <owl:DatatypeProperty rdf:ID="edition">
        <rdf:type rdf:resource="&owl;FunctionalProperty"/>
        <rdfs:domain rdf:resource="#infotainment"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </owl:DatatypeProperty>
    <owl:Class rdf:ID="geoenvironment"/>
    <geoenvironment rdf:ID="geoenvironment_1">
        <speedlimit rdf:datatype="&xsd;int">100</speedlimit>
        <slope rdf:datatype="&xsd;int">10</slope>
        <Sinuosity rdf:datatype="&xsd;int">60</Sinuosity>
    </geoenvironment>
    <geoenvironment rdf:ID="geoenvironment_2">
        <speedlimit rdf:datatype="&xsd;int">105</speedlimit>
        <slope rdf:datatype="&xsd;int">15</slope>
        <Sinuosity rdf:datatype="&xsd;int">10</Sinuosity>
    </geoenvironment>
    <geoenvironment rdf:ID="geoenvironment_3">
        <speedlimit rdf:datatype="&xsd;int">90</speedlimit>
        <slope rdf:datatype="&xsd;int">5</slope>
        <Sinuosity rdf:datatype="&xsd;int">30</Sinuosity>
    </geoenvironment>
```

Fig. 8. The OWL/XML file generated depicting the context

5 Conclusions and Future Work

An IoV system enabling interaction among vehicles in large scale has been presented in the paper. Each vehicle in the system abstracts and organizes the data it sensed, and shares them with other vehicles that have the common interests denoted using contexts. The sensed data are abstracted and organized using hierarchical context and content names. ICN technique is used to distribute the data and push-based data distribution mode is introduced considering the real-time characteristic of vehicle-related information. Depending on the characteristics of the information, the data can be encapsulated using different application layer protocols and transmitted in the network. At the network layer, the data packet can be forwarded in both opportunistic or storing and forwarding mode according to the network conditions and the abstracted characteristics of the content. The content abstraction and organization block of our system has been implemented and tested. The MediaSense can be provided as an open source platform providing standard sensor and actuator API.Currently we are implementing and testing other modules of the system. Security such as DoS attack is also our consideration of next step work.

Acknowledgements. Research funding from the European (FP7) MobiS Project is deeply appreciated.

References

1. Hossain, E., Chow, G., et al.: Vehicular Telematics over Heterogeneous Wireless Networks: A survey. Journal of Computer Communications 33(7), 775–793 (2010)
2. Yu, Y., Tandiono, C., Li, X., et al.: ICAN: Information-centric Context-aware Ad-hoc Network. In: ICNC 2014, Honolulu, Hawaii (February 2014)
3. Chen, W., Cai, S.: Ad hoc Peer-to-Peer Network Architecture for Vehicle Safety Communications. IEEE Communications Magazine 43(4) (April 2005)
4. Tsilopoulos, C., Xylomenos, G.: Supporting Diverse Traffic Types in Information Centric Networks. In: ACM SIGCOMM Workshop on Information-Centric Networking, Toronto, Canada (August 2011)
5. Leal, M., Röckl, M., Kloiber, B., et al.: Information-centric Opportunistic Data Dissemination in Vehicular Ad-hoc Networks. In: 13th International IEEE Annual Conference on Intelligent Transportation Systems, Madeira Island, Portugal, September 19-22 (2010)
6. Kanter, T., Forsstrom, S., Kardeby, V., et al.: MediaSense– an Internet of Things Platform for Scalable and Decentralized Context Sharing and Control. In: ICDT 2012 (2012)
7. McGuinness, D.L., van Harmelen, F.: OWL web ontology language overview. W3C recommendation 10.2004-03, 10 (2004)
8. Web ontology language, `http://www.w3.org/2001/sw/wiki/OWL`
9. Ahlgren, B., Dannewitz, C., Imbrenda, C., et al.: A Survey of Information-centric Networking. IEEE Communications Magazine (July 2012)
10. Jacobson, V., Smetters, D.K., et al.: Networking Named Content. In: 5th International Conference on Emerging Networking Experiments and Technologies (December 2009)
11. Li, F., Chen, F., Wu, J.: Fast Longest Prefix Name Lookup for Content-centric Network Forwarding. In: 8th ACM/IEEE Symposium on Architecture for Networking and Communications Systems (2012)
12. MediaSense open source platform, `http://www.mediasense.se`
13. Chen, H., Finin, T., Joshi, A., Kagal, L.: Intelligent Agents Meet the Semantic Web in Smart Spaces. IEEE Internet Computing 8(6), 69–79 (2004)