Web of Needs - A New Paradigm for E-Commerce

Florian Kleedorfer, Christina Maria Busch,
Gabriel Grill, Soheil Khosravipour,
Fabian Salcher & Alan Tus
Studio Smart Agent Technologies
Research Studios Austria Forschungsgesellschaft mbH
Thurngasse 8/2/16
1090 Vienna, Austria
http://sat.researchstudio.at
Email: {firstname.lastname}@researchstudio.at

Erich Gstrein
Smart Engine GmbH
Mooslackengasse 17
1190 Vienna, Austria
http://smartengine.at
Email: erich.gstrein@smartengine.at

***Abstract*—Structurally, the Web resembles a marketplace: Web clients download documents from servers just like customers buy goods from kiosks. Interestingly, the marketplace metaphor for the Web goes further: most Web users are in the role of the customer and the documents downloaded in most cases actually represent commercial offers. This second resemblance leads to a fundamental asymmetry: in general, commercial offers are represented as first-class objects on the Web. Consumer needs, however, the very notions that drive the interaction, only materialize in the form of search terms or browsing behaviour. In this paper, we present an approach for explicitly representing needs as first-class objects on the Web of data in a way that allows interacting with them.**

I. Introduction

E-commerce today can be seen as a huge marketplace dominated by market criers. Producers/providers are populating the web with services/items they assume to be useful to consumers. Although these assumptions might be based on intensive market analysis they are always out-dated by nature and thus cannot cover near-term trends or needs of small target groups. As a result consumers get lost in searching for appropriate items/services satisfying their needs and the merchants do not know if they are offering the right goods.

In the late nineties, research on agent based e-commerce marketplaces [1]–[4] became very popular for tackling these problems. The key characteristics of these solutions were that autonomous agents [5] - sometimes equipped with the ability of mobility [4] - populated virtual marketplaces [2], [6] where they would trade. According to Guttman([1]) trading consisted of the three stages of the Consumer Buying Behaviour framework: product brokering, merchant brokering and negotiation. Elaborate systems like MATE (Multi-Agent Trading Environment) [6] provided additional *purchasing agents* supporting the consumer/buyer both at the specification and the evaluation stages of the product and merchant brokering. Furthermore, the multi-attribute utility theory [7] formed one popular basis for implementing the negotiation process and fuzzy-logic [8] was used to find matching products within the merchant's archive. [6] Although showing impressive results these electronic marketplaces failed to be widely used in industry due to their complexity and proprietary characteristics.

In contrast, Web Of Needs (WON) is taking a more pragmatic approach by using and combining well-known Web technologies. Instead of stressing the concept of autonomous, mobile agents being responsible for trading on the consumer's or merchant's behalf, WON is tackling this challenge more as a process of structured Web document authoring (need description) and subsequent matching. Consumers and merchants without specialized computer skills will be provided with tools that help formulate needs explicitly in a machine-readable, semantically rich manner to be published on the Web. Smart matching services will find appropriate offers for published needs thus providing the starting-points for interaction or negotiation.

On a technical level a general and open infrastructure based on common Web technologies for resource transfer such as buying, selling, or renting is being implemented. The infrastructure we are aiming for is open in the sense that any interested party can take part in the resulting technical system by publishing services conforming to the protocols and description languages. This technical system, in principle, represents a global marketplace without the limitations of current marketplace solutions, which are limited to a single website, region, product type, or interaction type.

II. Problem Statement

Planning a trip or vacation can easily turn into a nightmare when special interests - not reflected in the standard catalogue offers - have to be taken into account. Such a situation might not only arise when some daredevils want to go for an extraordinary adventure trip but also when a family with a higher number of kids wants to spend their holidays at the seaside. Even if we reduce this planning task to only finding an appropriate accommodation, hours and hours are easily spent scanning offers, writing emails or conducting clarifying phone calls. On the other hand, the landlord providing accommodation is often confronted with a bulk of unanticipated special requests on site. In an even worse scenario, he is left to wonder why no guests are booking his accommodation at all.

The major problem underlying the scenario above is that the consumers are not able to express their needs or wishes in a an appropriate way so that a producers can find them and react. As a result, users are spending much time in browsing through offers created by merchants having tried to guess their needs. Personalization is a technology often applied in situations where users need guidance in navigating through large product archives. (Burke gives a good overview of the field that the authors have also contributed to. [9]–[11]) However, creating meaningful preference profiles often is not very successful when no or only few historical data points exist or even when only implicit feedback - such as purchase information - is available. Having booked a location does not imply automatically that someone likes it - it could well be that this was simply the least bad offer found or available. However, offering products based on these data might not be the best business strategy for producers to rely on.

Empowering consumers and producers to define their needs in appropriate ways will lead to a better quality of service as merchants are not forced any more to rely on second guesses or on outdated market analyses.

III. Our Approach

A. Overview

Our overall goal is to create a decentralized infrastructure that allows people to publish documents on the Web which make it possible to contact each other, and this should only happen if all parties involved have an interest in doing so. The said document may contain a description of a product or service required or offered, a description of a problem to be solved with the help of others, an invitation to social activities, or anything else users may think of. On this abstract level of description, the document can be said to represent an interest in or a need for some kind of interaction with others. Therefore, we refer to this document as a *need*. It is the central entity of the system we propose. Each need has a globally unique identifier and an *owner*, i.e., a person or other entity that creates and controls it. When need owners want to communicate with each other, a *connection* object is created for each need involved. The connection is the second important entity in our design.
For the interchangeable formulation of needs, a common modeling language is required along with a publishing mechanism. The resource description framework (RDF) [12] as a basic technology and the principles of linked data publication [13] are used for this purpose. When needs are published on the Web, independent matchmaking services crawl them (or are informed of them in other ways) and look for suitable matches. A protocol defines how these services inform the need owners of possible matches. Matching services are required to honor the description language specific to needs; in addition to that, they must also be able to discover matches between resources described in different vocabularies.

The proposed infrastructure is a network consisting of at least three different types of nodes: *owner applications*, *web of needs (WON) nodes*, and *matching services*, as shown in Figure 1.

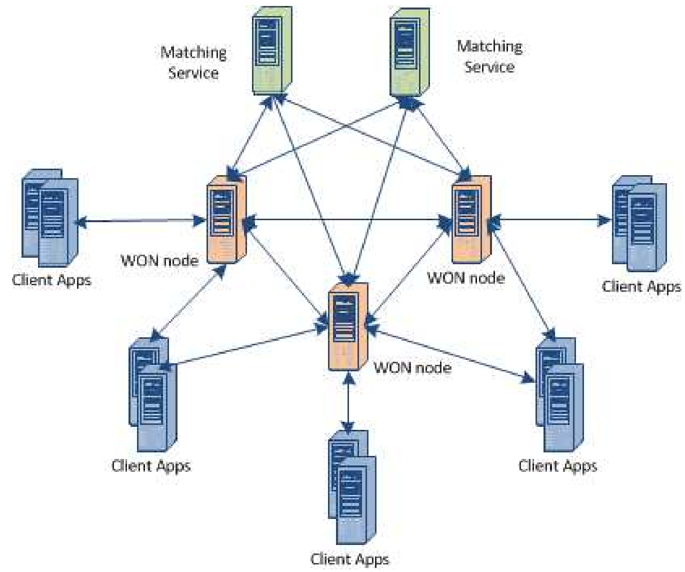

Fig. 1. Deployment diagram indicating types of nodes and communication paths. Needs are published as linked data on WON nodes, matching services crawl the data and report good matches; applications make the functionality available to users.

- *WON nodes*, or *nodes* for short, store needs, receive *hints* from matching services, and serve as communication relays between need owners. All needs stored on a WON node are published as linked data.
- *Matching services* constantly crawl the Web of needs, i.e., the part of the linked open data graph located on WON nodes. Whenever they find needs that satisfy each other's matching criteria, the respective need owners are given the appropriate information.
- *Owner applications* are applications that make use of the technology, either by direct interaction with users or by connecting to enterprise resource planning systems or other corporate software. They connect to one or more WON nodes to manage (create, update, delete) needs that are stored there and to communicate with other need owners.

For a more detailed overview of the system the interested reader is referred to a previous publication. [14]

B. Model

To exchange information about *needs* and describe the *need* itself we use an ontology. This ontology is used to publish the need as linked data as well as communicate about the need via the protocols described in more detail in the next chapter.

Since needs in our system are not restricted to a specific domain it is crucial to make the description of needs as open as possible. Therefore, the ontology explicitly allows for describing important details of the need using any other ontology. The ontology provides structures for constraints and meta data that are common to a many use cases and lower the bar for creating matching services that can produce high quality matchings; some of the meta data are simply required for the correct working of the protocol.

Before constructing our own ontology we considered work already done in this field. The main ontology in this field is Good Relations (GR) [15], which allows representing e-commerce processes, products, and company data. To evaluate the ontology for our purposes we defined 29 competency questions covering basic Web of needs functionality. With GR it is possible to cover 6 of them completely, 6 of them partly and it is not possible to answer 17 of them. Based on these results the decision was made to create our own ontology and import concepts from GR and other ontologies like Dublin Core (DC) [16] and WGS84 Geo Positioning (GEO) [17].

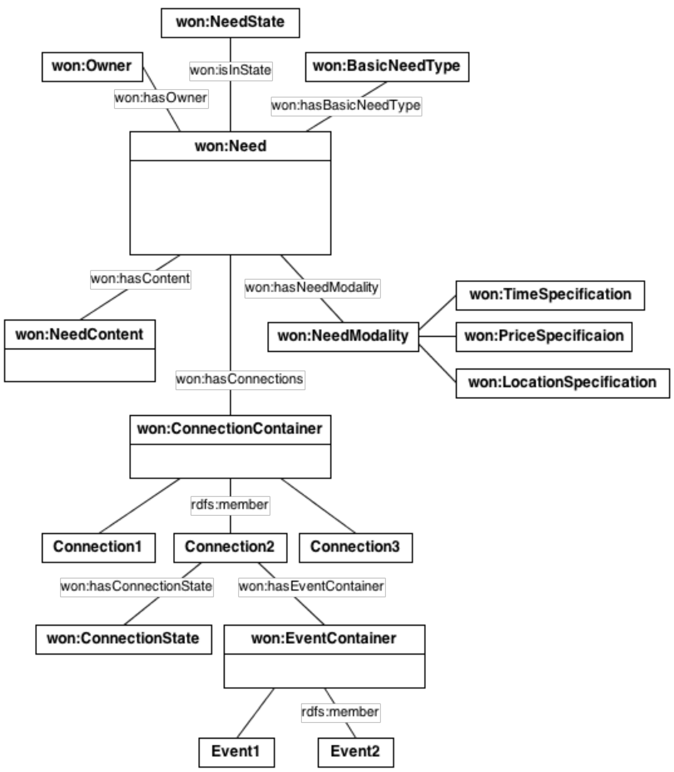

Fig. 2. A simplified model diagram.

Figure 2 represents a simplified model of the Web of needs ontology. Every need has a `NeedState`, `BasicNeedType`, `Owner`, `NeedContent` and one or more `NeedModality` and `ConnectionContainer` nodes.

The `NeedState` simply tells us whether the need is `ACTIVE` or `INACTIVE` since we can not interact with an inactive need.

Needs can be classified by the match counterparts they demand. E.g., if you exchange physical things there is always one part giving the thing away and one part receiving the thing or if you offer services there is one part offering the service and one part consuming the service. This is what we call need type. Basic need types are:

`SUPPLY` giving away items or supplying services,
`DEMAND` receiving a thing or consuming a service and
`DO_TOGETHER` perform some kind of activity together.

For each type, counterparts can be defined, which are allowed to be matched with the need. This is defined with the `allowsMatchWith` property. `SUPPLY` and `DEMAND` can be matched to one another and `DO_TOGETHER` can be matched only with itself. This gives us flexibility in defining new types of needs and influencing matcher behaviour declaratively.

Each need has an `owner` which is a description of a person or organization[1] or just an anonymized owner represented by a URI.

The need title and description, with some other features are described within the `NeedContent` node. It provides properties to describe the need with a simple description as a string or as a complex (or simple) ontology which is free to choose or construct. Additionally to support the matchers we provide a mechanism to describe physical objects in terms of physical dimensions like height, depth, width, and weight based on GR properties.

`NeedModality` structures specify the `PriceSpecification` (upper and lower limit, currency), `LocationSpecification` (latitude and longitude or region defined by the ISO 3166-1 and ISO 3166-2 codes) and `TimeSpecification` (start and end time, recurrence).

The `ConnectionContainer` is the part of the need that is gradually filled with events as the need lives in the system: Two[2] needs which interact with each other are represented by a `Connection`. Such a connection represents all events which can occur between two needs. At the moment such events are:

`HINT` received a notification from a matcher when a fitting need has been found,
`OPEN` received a request to establish a connection with another need and
`CLOSE` closing the connection between two needs.

This life cycle will be more closely defined in section III-C of this paper.

Figures 5 and 6 respectively represent a real life example of `SUPPLY` and `DEMAND` needs. They are written in Turtle [18].

C. Protocols

As we explained in the overview section, WON nodes are central points of communication in the web of needs. All components send and receive messages through WON nodes (see Fig. 1). In this section we describe the three protocols used to enable standardized communication between all the previously introduced components.

In our current approach we chose SOAP Web services to create an event-driven architecture [19], where all events that affect a need are propagated to the respective owner application. All needs and their connections are published to the Web as linked data. This enables crawlers to mine the data with ease and use it to match needs. All needs are identified by a unique URI, which can be used to access the RDF graphs.

Figure 3 shows how two parties can make a connection. In the example a customer (owner) informs (createNeed) a WON

[1]In this case we make use of the schema.org initiative.
[2]The ability to interact between more than two needs will be added later.

node of the need for a hotel. The WON node publishes the data to the Web. Meanwhile, a hotel manager (owner) sends the need to rent out a room to another WON node. Later on a matching service finds the data by crawling WON nodes and tries to match it with the data of other already crawled needs. If a match is found the respective WON nodes get notified by the matching service and inform (hint) the owners of the needs that a match has been found. After receiving the notification the owners can open a connection to each other. When both owners have sent an request to open the connection, text messages can be exchanged until one of the parties decides to close the connection.

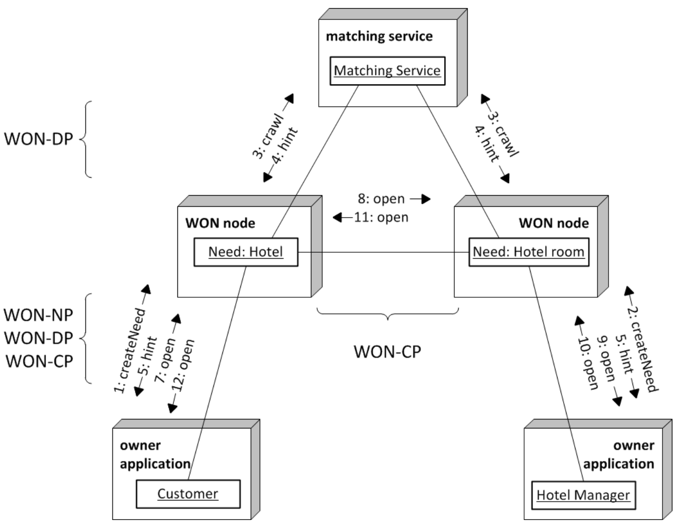

Fig. 3. Needs are created by owner applications on WON nodes, matching services crawl the data and report good matches.

In the following list we will describe the three protocols of the web of needs:

1) Discovery Protocol: The Discovery Protocol (`WON-DP`) consists only of one message called `hint`. If a matching service finds two matching needs, this message can be send to the respective WON nodes to indicate that a match has been found. The `hint` messages is then propagated to the respective owner application. A `hint` message consists of the URIs of the affected needs, a matching score, an URI of the matching service and an optional RDF graph. As shown in Figure 4, a `hint` message creates a connection in state `suggested` on the WON node. The URI of the created connection is supplied to the respective owner. In addition the WON node publishes the received information of the `hint` message to the Web, so that other matching services can use this information. This is reflected in the model as a `Event` of type `HINT` and the hint message is attached to it.

2) Need Management Protocol: The Need Management Protocol (`WON-NP`) consists of messages to create, activate and deactivate needs. A `createNeed` message is sent by an owner application to a WON node and consists of an URI of the owner application, an RDF graph describing the need and a flag, indicating if the need is activated. The passed URI is stored as a callback on the WON node and used to propagate messages to the owner application in the future. In our current implementation the described URI points to a Web Service Endpoint, implementing a well-defined interface. The `activate` and `deactivate` message are used to set the state of needs. A deactivated need doesn't receive any messages from WON nodes. Needs are usually set to this state after the needs has been fulfilled.

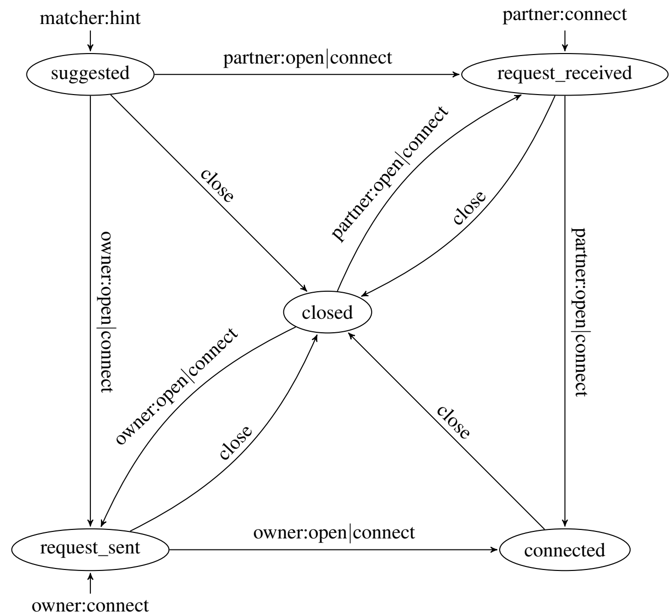

Fig. 4. Connection state machine

3) Connection Management Protocol: The Connection Management Protocol (`WON-CP`) specifies all messages needed to complete a transaction. Every sent `WON-CP` message to a WON node is published to the Web. Additionally an optional RDF graph can be passed along each `WON-CP` message to provide information on why it has been sent. Matching services can then use this information e.g. to improve their matching. The `connect` messages can be used to create and open a connection between two needs by providing their URIs. A connection URI is supplied to both owners after the message has been sent. The `open` message can be used to open an already existing connection. When a connection is established, strings can be exchanged between the connected owners of the needs via the `textMessage` message. A connection can be closed with the `close` message. Figure 4 shows the possible states of a connection and the `WON-CP` resp. `WON-DP` messages which trigger a transition. In the Figure the matching service is called "matcher" and the owner of a matched need is called "partner". Each time a new connection is created for a need, a new `Connection` node is added to the `ConnectionContainer` with an adequate `Event` of type `OPEN`. When a connection gets closed a or `CLOSE` event is added.

Note that, by a different naming scheme, we have used the terms Need Protocol (`WON-NP`), Matcher Protocol (`WON-MP`), and Owner Protocol (`WON-OP`). In this view, the protocols are classified by the interaction partners rather than the functionality. Figures 5 and 6 use this scheme for their pointers to Web service endpoints.

D. Implementation

The system we describe here is under development, the source code is available at github [20]. It is written in Java 7 with Spring [21], builds with Apache Maven [22] and is designed to run on a Java servlet container (developed on Apache Tomcat 7 [23]). The database backend is managed with Hibernate [24] on top of Spring Data - JPA [25], connected to either a HSQLDB or a PostgreSQL database [26], [27]. The Web layer is built with Spring WebMVC; the linked data functionality is based on the Jersey JAX-RS implementation [28], SOAP Web services use the JAX-WS reference implementation from the Metro project [29]. RDF handling is done via the Apache Jena RDF API [30].

The three basic types of nodes or services are built as follows:

- The *WON node* is realized as a Java Web application serving linked data and Web service endpoints for communicating with the other components.
- The *owner application* is realized as a Java Web application offering simple functionality (create and manage needs, connect needs with others, chatting) to end users.
- The *matching service* is realized as a linked data crawler, a slightly modified version of ldspider [31], [32], that periodically crawls the linked data on all known WON nodes. All documents (i.e. needs) that are found are sent to a SIREn(Solr) server [33], where for each document in the index the most similar ones are identified and if the match score exceeds a threshold, `hint` messages are sent to the matching needs.

IV. Future Work

Current development focuses on delivering a proof-of-concept that is available as open source and will be refined iteratively. We will provide an implementation of each of the required node types and run an online demo of each of them. Further work will provide solutions for authentication, security, and privacy, a more usable user interface, and a more scalable matching service. Moreover, we plan to research how users structure their needs cognitively so as to guide the design process further down the road.

Design and conduction of technical and psychological methods and experiments that can be used for evaluation of this project will be the ultimate steps of our research. The evaluation has to be done not only on different parts of the infrastructure, including need description, matching, user interface, etc. but also on the WON as a whole phenomenon. According to different psychological theories (e.g. [34]) the necessary conditions that have to be fulfilled for psychological growth, integrity, and well-being are specified by human needs. We will try to investigate the influence of application of WON, as a need satisfaction tool, on short-term and long-term psychological factors of users, which could lead us to new research directions.

V. Conclusion

We have motivated the creation of a decentralized Web based infrastructure for the management and satisfaction of human needs. We have given an overview of its design, leveraging the semantic Web and linked data technology stacks. Our proposal defines the overall shape of the system, and by doing so, it opens a range of new possibilities and questions which we have enumerated. The original contribution of this work consists of the motivation and the detailed description of model, protocols, and implementation.

Adding the concept of needs to the current *web of offers* might revolutionize e-commerce as we are used to today. Beside consumers and producers also technology providers will benefit from the presented approach, because many different services and/or tools such as matching services, need-crawler, need-definition and explanation assistance, etc. have to be provided. Based on that a new, semantics-based 'need-satisfaction' ecosystem evolves which adapts to market changes or specific/specialized requests of small target groups much faster and with lower costs. WON offers a wide field for research and development tasks where also small and innovative companies can play an important role. This is the major reason why a start-up SME such as Smart Engine is taking part in such an ambitious research project.

Acknowledgment

This work was supported by the Austrian Federal Ministry of Science and Research and the Austrian Research Promotion Agency (FFG) in the BRIDGE project *WIN - Web of Needs Infrastructure*.

References

[1] R. H. Guttman, A. G. Moukas, and P. Maes, "Agent-mediated electronic commerce: a survey," *Knowl. Eng. Rev.*, vol. 13, no. 2, pp. 147–159, Jul. 1998. [Online]. Available: http://dx.doi.org/10.1017/S0269888998002082

[2] J. C. Collis and L. C. Lee, "Building electronic marketplaces with the zeus agent tool-kit," in *Selected Papers from the First International Workshop on Agent Mediated Electronic Trading on Agent Mediated Electronic Commerce*, ser. AMET '98. London, UK, UK: Springer-Verlag, 1999, pp. 1–24. [Online]. Available: http://dl.acm.org/citation.cfm?id=648177.751311

[3] R. Sesseler and S. Albayrak, "Agent-based marketplaces for electronic commerce," in *International Conference on Artificial Intelligence, Las Vegas*, 2001.

[4] D. B. Lange and O. Mitsuru, *Programming and Deploying Java Mobile Agents Aglets*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1998.

[5] H. S. Nwana, "Software agents: An overview," *Knowledge Engineering Review*, vol. 11, pp. 205–244, 1996.

[6] M. J. Owen, L. Lee, G. Sewell, S. Steward, and D. Thomas, "Multi-agent trading environment," *BT Technology Journal*, vol. 17, pp. 33–43, 1999.

[7] H. Raiffa, *The art and science of negotiation*. Cambridge, Mass.: Harvard Univ. Press, 1982. [Online]. Available: http://gso.gbv.de/DB=2.1/CMD?ACT=SRCHA&SRT=YOP&IKT=1016&TRM=ppn+027388069&sourceid=fbw_bibsonomy

[8] L. A. Zadeh and J. Kacprzyk, *Fuzzy logic for the management of uncertainty*. John Wiley & Sons, Inc. New York, NY, USA, 1992.

[9] R. Burke, “Hybrid recommender systems: Survey and experiments,” *User modeling and user-adapted interaction*, vol. 12, no. 4, pp. 331–370, 2002.

[10] E. Gstrein, F. Kleedorfer, R. Mayer, C. Schmotzer, G. Widmer, O. Holle, and S. Miksch, “Adaptive personalization: A mulit-dimensional approach to boosting a large scale mobile music portal,” in *Integration of Music in Multimedia Applications*, A. Kohl, Ed., 2005, pp. 1–8.

[11] easyrec website. Research Studio SAT. [Online]. Available: http://easyrec.org

[12] F. Manola and E. Miller, *RDF Primer*, 2004. [Online]. Available: http://www.w3.org/TR/rdf-primer/

[13] W3C. [Online]. Available: http://www.w3.org/standards/semanticweb/data

[14] F. Kleedorfer and C. M. Busch, “Beyond data: Building a web of needs,” in *Proceedings of the WWW2013 Workshop on Linked Data on the Web (LDOW 2013)*, Rio de Janeiro, Brazil, May 2013, to appear.

[15] M. Hepp, “Goodrelations: An ontology for describing products and services offers on the web,” in *Proc. of the 16th Int. Conf. on Knowledge Engineering and Knowledge Management (EKAW2008)*, ser. Springer LNCS, vol. 5268, Acitrezza, Italy, September 2008, pp. 332–347.

[16] D. Hillman, *Using Dublin Core*, 2005. [Online]. Available: http://dublincore.org/documents/usageguide/

[17] D. Brickley, *Basic Geo (WGS84 lat/long) Vocabulary*, 2003. [Online]. Available: http://www.w3.org/2003/01/geo/

[18] D. Beckett and T. Berners-Lee, *Turtle - Terse RDF Triple Language*, W3C, 2011. [Online]. Available: http://www.w3.org/TeamSubmission/turtle/

[19] B. M. Michelson, “Event-driven architecture overview,” *Patricia Seybold Group*, vol. 2, 2006.

[20] Web of needs github project. Research Studio SAT. [Online]. Available: http://github.com/researchstudio-sat/webofneeds/

[21] R. Johnson *et al.*, *Spring Java Application Framework Reference Documentation*, 2012. [Online]. Available: http://static.springsource.org/spring/docs/3.1.x/spring-framework-reference/pdf/spring-framework-reference.pdf

[22] Maven developer centre. [Online]. Available: https://maven.apache.org/developers/index.html

[23] Apache tomcat 7 documentation index. Apache Software Foundation. [Online]. Available: http://tomcat.apache.org/tomcat-7.0-doc/index.html

[24] *Hibernate Core Reference Manual*, Red Hat, Inc, 2013. [Online]. Available: http://docs.jboss.org/hibernate/orm/4.1/manual/en-US/html/

[25] O. Gierke, *Spring Data JPA - Reference Documentation*, 2012. [Online]. Available: http://static.springsource.org/spring-data/data-jpa/docs/current/reference/pdf/spring-data-jpa-reference.pdf

[26] B. Simpson and F. Toussi, *HyperSQL User Guide*, 2013. [Online]. Available: http://hsqldb.org/doc/2.0/util-guide/util-guide.pdf

[27] *PostgreSQL 9.1.9 Documentation*, The PostgreSQL Global Development Group, 2012.

[28] Jersey 2.0 user guide. Java.net. [Online]. Available: https://jersey.java.net/nonav/documentation/latest/index.html

[29] Jax-ws release documentation. Java.net. [Online]. Available: https://jax-ws.java.net/2.2.7/docs/

[30] The core rdf api. Apache Software Foundation. [Online]. Available: http://jena.apache.org/documentation/rdf/

[31] R. Isele, J. Umbrich, C. Bizer, and A. Harth, “LDSpider: An open-source crawling framework for the web of linked data,” in *Proceedings of 9th International Semantic Web Conference (ISWC 2010) Posters and Demos*, 2010. [Online]. Available: http://iswc2010.semanticweb.org/pdf/495.pdf

[32] Ldspider4won github project. Research Studio SAT. [Online]. Available: https://github.com/researchstudio-sat/ldspider4won

[33] R. Delbru, “Siren: entity retrieval system for the web of data,” in *Proceedings of the Third BCS-IRSG conference on Future Directions in Information Access*, ser. FDIA’09. Swinton, UK, UK: British Computer Society, 2009, pp. 29–35. [Online]. Available: http://dl.acm.org/citation.cfm?id=2227296.2227302

[34] E. L. Deci and R. M. Ryan, “The ”what” and ”why” of goal pursuits: Human needs and the self-determination of behavior,” *Psychological Inquiry*, vol. 11, no. 4, pp. 227–268, 2000. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1207/S15327965PLI1104_01

APPENDIX

```
@base <http://example.com/won/> .
@prefix won: <http://purl.org/webofneeds/
    model#> .

:ld/resource/need/3
  a won:Need ;
  won:hasBasicNeedType won:Demand ;
  won:isInState won:Active ;
  won:hasContent
    [ a won:NeedContent ;
      dc:title "Hotel" ;
      won:hasTextDescription "I am
          looking for a 3 or 4 star hotel
          on Hvar, Croatia between the
          dates of 15.7. and 25.7."
    ] ;
  won:hasNeedModality
    [ a won:NeedModality ;
      won:hasLocationSpecification
        [ a geo:Point ;
          geo:latitude "43.175" ;
          geo:longitude "16.652"
        ] ;
      won:hasPriceSpecification
        [ a won:PriceSpecification ;
          won:hasCurrency "EUR" ;
          won:hasUpperPriceLimit "1000.0"
        ]
    ] ;
  won:needCreationDate "2013-05-144T11
      :19:53.053" ;
  won:hasConnections
    :ld/resource/need/3/connections/ ;
  won:needProtocolEndpoint
    :protocol/need ;
  won:ownerProtocolEndpoint
    <protocol/owner ;
  won:matcherProtocolEndpoint
    <protocol/matcher .

:ld/resource/need/3/connections/
    a ldp:Container .
```

Fig. 5. An example of a need, written in Turtle. Namespace prefix declarations of commonly known namespaces are omitted for brevity.

```
@base <http://example.com/won/> .
@prefix won: <http://purl.org/webofneeds/
   model#> .

:ld/resource/need/4
  a won:Need ;
  won:isInState won:Active ;
  won:hasBasicNeedType won:Supply ;
  won:hasContent
    [ a won:NeedContent ;
      dc:title "Hotel room" ;
      won:hasTextDescription "We are
         renting out rooms in Hotel 'Hvar
         ' during the summer."
    ] ;
  won:hasNeedModality
    [ a won:NeedModality ;
      won:hasLocationSpecification
        [ a geo:Point ;
          geo:latitude "43.175" ;
          geo:longitude "16.652"
        ] ;
      won:hasPriceSpecification
        [ a won:PriceSpecification ;
          won:hasCurrency "EUR" ;
          won:hasLowerPriceLimit "50.0" ;
          won:hasUpperPriceLimit "100.0"
        ] ;
      won:hasTimeSpecification
        [ a won:TimeSpecification ;
          won:hasEndTime "2013-08-01T12
             -00-00" ;
          won:hasStartTime "2013-06-01T12
             -00-00"
        ]
    ] ;
  won:needCreationDate "2013-05-144T12
     :07:40.040" ;
  won:hasConnections
    :ld/resource/need/4/connections/ ;
  won:matcherProtocolEndpoint
    :protocol/matcher ;
  won:needProtocolEndpoint
    :protocol/need ;
  won:ownerProtocolEndpoint
    :protocol/owner .

:ld/resource/need/4/connections/
  a ldp:Container .
```

Fig. 6. An example of an offer, written in Turtle. Namespace prefix declarations of commonly known namespaces are omitted for brevity.