Information Services & Use 29 (2009) 73–80
DOI 10.3233/ISU-2009-0600
IOS Press

Second keynote

The Semantic Web. Enabling innovative approaches for handling information and services

Prof. Dr. Rudi Studer
Immediate Past President, Semantic Web Science Association, Karlsruhe Institute of Technology & FZI Research Center for Information Technology

Ladies and Gentlemen, I would now rather move into a more technical presentation compared to the first keynote speaker. So the basic idea is to outline: What kind of methods we have around in the Semantic Web area and how these methods could be used for developing, what I called "Innovative Approaches for Handling Information and Services".

What's the outline of my talk? Well, I would like to start with the presentation: What are the basic ingredients of that Semantic Web vision. Then I would like to address more application oriented aspects and show you some scenarios, where you can see, how these methods are applied in innovative ways. One is: How you can use semantically enhanced Wiki environments to provide the content you need in Semantic Web applications. The third aspect will address the application you are all experts in: how you can apply Semantic Web standards in order to describe licence information, licence issues and then exploit that kind of models in order to access publications and pictures or whatever you are interesting in the end. And I would like to conclude with some personal vision over what I think is what some people call "Web 3.0", some people still call it "Semantic Web", some people call it "Enhanced Web 2.0". So there are a lot of notions around. I will use the term "Web 3.0" in order to indicate what I see as a trend and what kind of technologies really come together to realize that vision. Ok, let me start with describing the idea of the Semantic Web. Well, we are all for familiar with what I call here the "Classical Web". What is the basic idea? We have a lot of content and I stress here the semi-structured content, a lot of documents, maybe some documents describe XML and these kind of standards. And the basic idea is that content is provided for human consumption. So that you as a reader browse around in the web and you collect the information from these web resources in order to get informed about the information, you are interested in. The main problem we have, when you think of automatising some of these processes is, that when you think of application-systems or computers, that should get hold of the content of these kinds of documents. It's very difficult to grasp automatically the meaning of that content. And what is the Semantic Web vision about how to get rid to some extent of that problem. So that computers are able to really get hold of some aspects of the content of your documents. And the third ingredient of the classical web is that keyword-based search that is provided by all these nice companies and tools you are all familiar with. And the basic problem you

see there is that rather popular example when you ask for Jaguar, as one example, you might get information from all different kind of applications domains. So some people might be interested in buying a nice car on Saturday morning. So when they talk about Jaguar, when they want to get information about Jaguar, they want to know, what kind of cars are around, what kind of models are offered – that kind of thing. But when you rather go for family excursion on Sunday afternoon, you might rather be interested in getting more information about some kind of animals having four legs and that kind of stuff. And that is the classical problem with our standard version of the web we have around, that all these systems are not really able to grasp the context, we are working in. So they do not know, whether we are on the business trip or want to sell – want to buy some car or whether we are on a family trip and want to have some entertainment for our children. So they lack that context information that then might give a clue to handle that problem. That Jaguar comes with different meanings and the system should know what meaning is important for you in this specific situation. And that is, what you are currently not able to really handle with the classical approach. So what is the basic idea of the Semantic Web? To move on one step to go from that semi-structured information to information, that is structured and what is especially important, that it is linked to each other. So to come up with structures, where you link all that information, that is around on the web and the second basic ingredient is, that you provide on top formal models, that are called "Ontologies" in our community – that provide the meaning for these data and the relationships, they are embedded in. And by using these two basic ingredients, that link data formats and I will come to that in a few minutes and these ontologies on top of that, you are to some extent able to provide meaning in a way, that application systems or computers are able to understand the meaning and then to process that meaning. And in that way you then are able to come up with innovative applications to better integrate information, that comes from different sources: what is the meaning, that comes from source "X" and what is the meaning, that comes form source "Y". So that you do not compare different meanings in your application, that integrates information from these two sources. What we also have around is the World Wide Web consortium that is heavily involved in developing and then standardizing, languages and formats in order to structure these data and in order to structure these models, so that you rely all on World Wide agreed standards – so that you are really able to exchange that information between different application systems.

Well, let me briefly discuss two of these basic ingredients that are very important for the Semantic Web. One is the basic data model, it's called "RDF", Resource Description Framework and that rather very simple data model is the "Basic layer", we are using for the Semantic Web approach. It's rather generic, so you are able to describe all kind of data using RDF. And that is its power, so we are not tailored to specific application domains, but you can use that in different applications. When you look into the formal basis of that model, you could say well, that is one of these well known graph-based models with some characteristics, but I will not discuss them today. And the strength of these graph-based models is that you are really able to support all these integration notions. I will have a few examples, indicating that. So, what are the basic modelling primitives of your RDF model? First we have resources – and that might be everything you are interested in, that might be my institute AIFB, so I have an institution, that might be my hometown Karlsruhe, that might be a book, you are publishing, that might be an author writing a book. So everything might be a resource. So that is a very generic term. The second basic ingredient is a property and that is used to link two resources to each other. So in my trivial example we have one property located in and that relates that institute resource AIFB to that resource "The city of Karlsruhe". You might also think of other properties like "written by" when you link a resource of a book with the resource of an author, for example. And that ends up in what is called "triples", because you always

have two resources linked by one property. So that is the basic structure of that model and by specifying assertions, we specify these kind of triples relating two resources via such a property. Why might that be interesting for combining information that comes from different sources? For example, we see here our first example, a little bit extended, so AIFB is located in Karlsruhe, Rudi Studer works at AIFB, we organized the International Semantic Web Conference, ISWC2008 in Karlsruhe. So ISWC2008 is also located in Karlsruhe and I had the pleasure to be the local organizer of that conference. So there are four triples that are related to that university conference researcher domain. On the other hand you might have another resource, that describes cities in Germany and Karlsruhe is one of these cities. So you know well, Karlsruhe is also located somewhere. It's located in Germany, in a country. It has some kind of population – 280,000 people living there and we have some information about the square meters that are covered by Karlsruhe. So you might have a different resource that is oriented towards providing information about cities. Then you might find a third resource providing information about companies and so on and so on. So find all these kind of resources around. And what is now a very nice feature of that triple based structure of the RDF model? That you can now combine these triples you have made and you see one shared resource between the university domain on the one hand and the city domain on the other hand i.e., that the resource Karlsruhe that is involved in both of these domains. And based on that standard triple structure, you can than integrate that information. So you integrate the information about Karlsruhe, with aspect of the city domain, with the information about Karlsruhe related to the university that is located in Karlsruhe. And in that way you can build up bigger and bigger models that are able to combine information from different kind of resources. And that is what Tim Berners-Lee coined as a term: the web of data. We have all these triples around and depending on the kind of application, you want to develop, you combine these triples in a very flexible way. So that is the basic data model that is that flexible and that primitive, that you can really capture all kind of information there. The second ingredients are these ontologies. Please be aware, we do not talk about the term ontology in the philosophical sense. We talk about the meaning of ontologies in the area of informatics and computer science and that is a rather technical term. We talk about models. And here I have a quote that definition given by Uschold more than ten years ago. An ontology is a shared understanding of some domain of interest. So here you see two aspects that are important for ontologies. One is the relation to some domain of interest. In my examples we talk about universities, we talk about organizations. You might also talk about libraries, publications – whatever you are interested in. And the second important aspect is that you talk about a shared understanding. And that means a group of people or a community has to agree on the vocabulary you are putting into these ontologies. Otherwise you can not share them between different applications, because people have to use this model and when you talk about, what does it mean to be an organization, you should have a consensus. You need an agreement. So that notion of a shared understanding is rather important. Again, when you think of a rather simple model of ontologies, what are the modelling primitives, you typically find in these ontologies? You find on the one hand a concept, like – my example, a person or a professor or an organization and these concepts are embedded in a generalisation hierarchy. So that you know, when you talk about a professor, such a professor has all characteristics a person has as well and in addition a professor has some additional characteristics. Or when you talk about a research institute, it has all characteristics of an organization, but it has some additional characteristics. So that's the meaning of these hierarchical structures, generalisation structures we talk about in these modelling terms. And as a second basic primitive you have these relations that are able to relate these different concepts to each other. So here I have that simple example that you have a

"Works-at" relationship between persons on the one hand and organizations on the other hand. So that is that concept level, where you talk about these general notions that are relevant for the application domain. Then you might instantiate your model with instances that are relevant in your application domain. So when you talk about the university domain, you might have some professors like myself, you might have some institutes like my home institute, but you could also do that in your publication domain talking about specific books, talking about specific companies that are publishers or talking about specific persons, that are authors of some books. So then you can talk about these specific persons, specific institutions by exploiting the vocabulary, that is provided by these ontologies. In my simple example you now know that Rudi Studer is an instance of professor and implicitly you know that Rudi Studer is also a person, so I have all characteristics of a person and I'm related to that research institute, instance AIFB, by exploiting that "Works-at" relationship. So that is that instance level. You could also see – however that is not showing up in my simple examples – by providing additional relationships on top at that basic model – that you, for example, might derive further implicit information. So when I talk about, that Rudi Studer works at AIFB, you can derive from that, the affiliation of Rudi Studer is AIFB and indirectly university of Karlsruhe, because you know AIFB is an institute at university of Karlsruhe. So using this kind of derivation mechanisms, you can derive more information based on these models. Why can you do these derivations? – Because these ontologies come in logical languages, that do not only have a well-defined syntax, but also comes with a well-defined semantics. And these semantics then describe what kind of inference step you are allowed to do in the end. So that is the second step in order to come up with these two ingredients for the Semantic Web. The basic data model RDF and then these ontologies on top of RDF that provide the meaning for these terms that are used in these models. Ok, so that is a very short introduction, what is the Semantic Web about. Let me now move into two application scenarios, one addressing the question: Where do all these triple based statements come from? Someone has to provide these triples. Or you need some automatic mechanism to generate these triples in order to populate the models of your application domain. So, that is one basic question. And one approach is to use these nowadays very popular Wiki environments with a little bit of semantics and use these Web 2.0 approaches to generate these instances for your models. What you see in this example is that Semantic MediaWiki system. MediaWiki is that software system that is, for example, running Wikipedia. And I assume all of you are using Wikipedia at least once a day, for example. So that approach is used as a basic software platform and then, in the end, we have extended that basic software with two extensions in order to cover specific semantic aspects. And what you see here is a development that has been initiated at my institute and it's also further developed at related institutes we have around in Karlsruhe. And if you are interested in, just go to the website, it's an open source software. You can just download and use it. And hundreds of people all around the world are using it. So what is the idea? You are all familiar with Wikipedia and these articles – here you find an article about Karlsruhe. Then you can get information about: Karlsruhe is located in Baden Württemberg, one of the local states in Germany. It has some number of inhabitants and all these things you get informed about a city. And as a human reader then you know all these facts. Well, when I ask you later, well, in which local country Karlsruhe is located, you would answer: Well, I have learned, that it is located in Baden Württemberg. So for human readers it is very easy to grasp these facts. But for computers it is very difficult to interpret this text and get hold of these facts that are somehow hidden in that text format. And what we have developed as extensions are two types of extensions. We just used that Wiki syntax and maybe some of you are familiar with all these square brackets you use in Wiki environments, but

do not worry about that. We just had put in two additional modelling primitives. One is, that you can specify explicitly such a property in that RDF model. So we have a resource Karlsruhe, because we are talking about Karlsruhe on that page. You want to indicate, that it is located in Baden Württemberg and then you specify that property, relating that resource Karlsruhe with that resource Baden Württemberg. And that is done with that, in German "liegt in" or "located in" relationship. I'm showing it in blue here. And in the same way you can relate Karlsruhe to other descriptive elements, like the number of inhabitants, like we have done it here. And by using that slightly extended syntax, the software in the background is able to extract these facts from the descriptive text and from that you generate, what we call "Fact box", where you now see three facts about Karlsruhe. One is the number of inhabitants that are living there. A second aspect is the number of square-meters that are covered by Karlsruhe and the third fact is, where in Germany Karlsruhe is located – in Baden Württemberg, for example. And since we are then using RDF triples, you can combine that with information about the University of Karlsruhe, about research projects there, so in the same way as I have indicated that a few minutes ago.

So that is the basic way of providing in a Web 2.0 way these statements about your application domain. What is then the benefit? What I have indicated here, you can specify, what we call "individual queries" – what is that about? When you browse through Wikipedia, you are always amazed by the number of summary pages you find there. You find lists about German cities, the biggest German cities and all kind of aspects. You find lists about the most popular films that had been produced in 2008 and all that kind of stuff. And how is that content produced? Well, somewhere in the world, there is one person sitting in the night in front of the PC and generating these summary pages and that has two deficiencies: One is: a lot of effort is involved, because that is done manually. Second: you have the consistency problem, because as soon as I change the number of inhabitants on the homepage of Karlsruhe, you should rather change the number of inhabitants on the summary page as well and maybe we rank the biggest cities in Germany according to that modification. And it is obvious, you will end up in inconsistency, because you changed the basic fact, but you forget to change it on all these summary pages. What we are able to do in the background, is taking these facts that we have generated, provide some mechanism that is done in database-like way and generate these summary pages in an automatic way. So you get rid of the manual effort and you get rid of the inconsistency problem, because all these pages are generated and therefore when you changed the basic fact, your summary page adapts that change and therefore it is consistent with the basic fact. So that is our second very big advantage of using these kinds of technologies in the background. Ok, when you think of what are these Wiki environments good for in the end, you can now come up with all different kind of applications, exploiting all these mashup techniques that have been become popular in the last three to four years, I would say. So you still rely on that Wiki based collaborative content editing. So everyone is allowed to edit a page about Karlsruhe. So that is the collective content provisioning process. By using our slightly enhanced semantic basic software, you are then able to provide these basic facts about the city, the university, whatever and can query that formally specified content and then finally use these mashup techniques like, for example, Google Maps, so that you can embed the information, you have just described in the Wiki system, in these Google Map environments. So you see here one example that is relating on pages about Semantic Web conferences and the location, where they have been organized with that Google Map environment. So then click to Karlsruhe and then you see well the International Semantic Web Conference 2008 was organized in Karlsruhe. So in that way you are very flexible in combing these mashup techniques that are provided by all these big companies – nowadays with the Wiki content you have edited in that collaborative way. And that is a very nice way of putting in these semantic applications in the standard environments you

are working with in your daily life anyway. Let me address a second application domain license information that might be directly related to what you are working in your company as well. And how can we exploit Semantic Web techniques in the background to come up with more innovative applications there? I use as a starting point creative commons, that is one of the very popular models for providing open access to content. And just as a side-remark, when you go to that web portal about creative commons, they are using our Semantic MediaWiki system in the background, so you can play around with that enhancement of the Wiki system as well. And what you then might be interested in are things like specific books or publications, and you might be interested in what is the license model that is attached to that. Is, for example, commercial usage allowed or forbidden, are you allowed to modify what you are getting hold of, so when you download some kind of picture, are you allowed to modify it and use it in your environment again. What kind of attribution you have to specify, to address all these kinds of aspects that are related to licence aspects – and let me show you that already some Semantic Web basic approaches are around in your application domain. You might refer to these creative commons rights expressing language that is, I would say, a rather lightweight ontology, but it provides the basic ingredients of such an ontology. So it models what is the work, it models what is the licence and it relates a work, for example, to the title or to the attribution and the licence, for example, to what it permits or what it requires. So that are relevant aspects, with aspect to your licence issues. And now let's see, how can we use that kind of model in order to describe a content you find on the web and clearly in order to use that on the web. You have to encode all that licence information in these basic semantic web standards and again you are able to encode that in RDF and embed it, for example, in the standard HTML format. But you can also use other multimedia formats and embed that RDF into these media formats. So here you see a very concrete example describing your work. When you read that web page, it's a presentation given by Even Hermann on the World Wide Web Consortium and he was talking about, what is the Semantic Web. And there you find some licence aspects being described on that page. You find well, it's related to 3.0 licence according to creative commons and it's attributed to the World Wide Web Consortium. So that are two aspects that describe, what is the legal or the licence environment, that presentation is embedded in. So that is the presentation on the web page for human consumption. You read that description, then you know, what it is all about. Do not worry about the syntax that is done by the machines in the background. But just that you aware, you can embed that formal description on these licence issues into your standard HTML format. And in the end, when you encode that you come up with that triple structure again, that you are now familiar with: so that presentation has a specific licence, that's 3.0 licence and it was attributed to the World Wide Web Consortium. So that is the basic licence information that is associated with that presentation. So we have some small basic collection of triples, describing that presentation. Well, but then the question: What does it mean to relay to that licence 3.0? Therefore we also describe what is that licence about, ok, and then you will see that specific aspects are related to that specific licence provided by that creative commons approach in the end. So again you could encode that in your RDF model, do not worry again about the syntax. But what you specified there: well it permits reproduction, it prohibits derivative works and it requires some kind of attribution. So you know these three things are important, when you talk about that specific licence, reproduction, derivative works and attribution are three aspects you have to think about. Ok, now we know, we have that presentation provided by Even Hermann, we have that description of that licence 3.0 and again, you can just combine it. So what you can then do is relate that licence information with the presentation information and again you have now more combined information around. When you want to look at the presentation of Even Hermann, then think about well: Am I allowed to reuse it, take two of his slides and put that into my presentation? You can combine that information and that it's what is

provided in the end. You can then exploit that for search – in the end, you specify for example, your licence requirements, what kind of rights or obligations you would like to allow or to disallow in the end and then you can exploit the information, that is already around. So when you go to Flickr, you find all that kind of information, for example, around, when you are interested in some kind of pictures. And then in the background you can match what you specify in your query with the description that is associated with your work, like a picture, a book or whatever and then you can see, what is qualifying. Here you see a simple example from our International Semantic Web Conference 2008, when you go to Flickr and you search for pictures about the International Semantic Web Conference, where my person is shown. And at the top you specify, what are the licence aspects, you want to address in the end. So I want to be allowed to use it for commercial purposes and I want to modify and adapt it. You can specify these requirements and then one picture qualifies. And then you know what you are allowed to do with that picture. Ok, let me address the last aspect: What is the trend towards Semantic Web 3.0. We started from Web 1.0, where you have this huge amount of semi structured content, that is meant for human consumption. What is the characteristic of Web 2.0 or Flickr and these other kind of popular websites? Well, they are characterized by collaboration and what is now called "Prosumers". So you have one person, that produces content and that consumes content. So that is that term, prosumer – that is very important for Web 2.0 aspects. When we then move on to I what would call Web 3.0, you still rely on that social web notions. So all that massive collaboration is around, you put in these ingredients of the Semantic Web that especially enhance your ability to provide data integration facilities. And what I see as a third aspect is, what some people called "The Service Web", that you also integrate these services, for example, showed in my example about these mashup environments based on our Semantic MediaWiki system.

What can you do by combining these three aspects: social web, Semantic Web and Service Web? Well, in the first place you can connect people to each other, done on these social networks or when you move into a business environment like competence management in companies. Or you can connect information by using all these Wiki's environments, where you relate information to each other or these social tagging approaches that relate different kind of information to each other. Or that information integration approach, I have just illustrated in my presentation several times. Then you can derive further knowledge, by, for example, using such a reasoning mechanisms in the background or by exploiting information extraction techniques that rely on nature language processing techniques and combine it with machine learning, for example. And then you can share knowledge by providing semantic search approaches and relying on these well defined standards, provided by the W3C that are used on a world wide level. So that you can easily exchange your ontology models with your friends or your partner company and also all these instance information you can easily exchange between these different environments. So that is from my point of view, what I would coin as the Web 3.0 approach. Ok, let me conclude with a brief summary: What is the take-home-message for you, for today? What should we have learned about Semantic Web and Web 3.0?

Well, Semantic Web provides well agreed standards to publish and integrate structured information. And the main advantage is that you provide precise semantics. So you are really able to integrate and process that information by machines and not only read them by human readers. You have seen that enhanced Wiki environments with these semantic techniques provides a very smooth way of providing the semantic content you are interested in the end. So you do not have to learn all these logical languages. You just rely on your Wiki environment, hack in the information you hack in anyway and in the background you generate that semantic information. We have seen that there are already first approaches around that rely on these semantic technologies. So in your domain you do not have to start from scratch.

You can rely on existing models in that context and what I see as a vision coming up in the next five years – I would say that combination of the Social Web, the Semantic Web and the Service Web. And that is what we would see in 2012, 2015 as the Web 3.0. Ok, thank you very much for your attention. In case you have further questions, I would be glad to answer them. Thank you.

References

[1] P. Hitzler, M. Krötzsch and S. Rudolph, *Foundations of Semantic Web Technologies*, CRC Press, 2009.
[2] S. Staab and R. Studer (eds), *Handbook on Ontologies*, 2nd edn, Springer-Verlag, 2009.