The WASP Framework – Bridging the Gap between the Web of Systems, the Web of Services and the Web of Semantics with Agent Technology

Thomas Biskup[1]
Jorge Marx Gómez[2]
[1]QuinScape GmbH
Thomasstr. 1, 44135 Dortmund
E-Mail: thomas.biskup@quinscape.de
[2]TU-Clausthal, Institute of Computer Science
Julius-Albert-Straße 4, 38678 Clausthal-Zellerfeld
E-Mail: gomez@informatik.tu-clausthal.de

Abstract

Currently we face a major gap between the reality of the web – a disjoined and tangled mass of loosely coupled information resources – and the vision for the web – a tightly integrated and openly structured information network with machine-readable data that allows autonomous agencies to create new applications empowered by this wealth of information. Current research shows that we can hope to achieve this goal, but there are many obstacles left to be mastered. We propose a framework to allow researchers and developers to choose the level of detail, the type of technologies and the extend of computing power they want to utilize for their proposed solutions. We focus on a flexible abstraction layer, pattern-oriented architecture and open interfaces to build on the successful foundations of the web: ease of use, flexibility and almost unlimited expression power. Agents are the central paradigm for software development using this architecture.

1 Introduction: The Web – Now and Then to Mobile Agents

The Semantic Web (see [3] and [4]) is a valuable vision pushed by the World Wide Web Consortium (W3C), and supported by vast research efforts, to build the future foundation for a true information society.

Nonetheless progress is slow and even if research would yield results at much greater speed (which does not seem reasonable since the open questions are truly awe-inspiring), the results still need to be implemented. Current research hints at much more expressive and thus also more powerful means to represent data and information – but the price is added complexity required to build the representations.

The World Wide Web was successful because people basically overnight were enabled to share information – with simple technology. This allowed for the enormous growth in information resources we now face and this pattern most likely should be reproduced to guarantee the further growth of the Web (see [5]).

1.1 The Web of Systems

Currently the World Wide Web is the largest information system ever build by humans – and it probably is also one of the least structured information systems build. There are billions of web pages (not counting other resources like images, videos, sounds, CGI interfaces to large databases and more) and almost none of them are structured in a standardized way. These pages mostly are build with HTML and only loosely coupled – links lead into oblivion as often as they do not. And even existing links do not provide much semantic information (e.g. what is the meaning of a specific link except "someone thought that two information resources should be connected in some way"). Most information is presented in a way that allows humans to use it – although access to this information usually is a problem because it becomes harder and harder to find the few tidbits of information in the existing mess of data.

Thus we argue that we need to find ways to evolve from the current World Wide Web (a Web of Systems – so named because there are many individual systems that usually are only connected by the simplest means, namely hyperlinks) to something more.

It would be foolish and dangerous to try too much at once. At the same time it would be as foolish and dangerous to create artificial boundaries and introduce building blocks that limit our power of expressiveness. Thus we propose to search for architectures and frameworks that support slow evolution without limiting the final goal. We find practical examples that support the viability of this approach: modular programming has spawned object oriented programming to be able to control complexity with more natural concepts. For certain problem areas agent-oriented systems have been discovered to be an immensely powerful and very natural concept for defining solutions (see [10]). Now the industry momentum offers a huge chance to solve one of the basic problems of agent societies: communication by web services promises to do away with the artificial system boundaries currently inhibiting large-scale distributed autonomous agent systems.

1.2 The Web of Services

Web Services (see [9] and [14]) – while surrounded by a lot of hype – for the first time in ages offer a standard means to communicate between disparate systems and applications with absolute disregard for programming languages, computer hardware and system-specific communication protocols. Based on XML (extensible markup language, see [6]) this new and exciting standard promises a new way of defining interfaces – without stickling to implementation details and basic technical questions. Together with HTTP (Hypertext Transfer Protocol, see [15]) and SOAP (Simple Object Access Protocol, see [26]) as protocols we face an enormous opportunity to bring together previously separated building blocks for the next generation internet. XML is the unifying data representation standard that could be used to basically encode any kind of information. HTTP and SOAP are simple yet flexible protocols that allow a system-independent communication between heterogeneous systems. While it currently is pretty difficult to connect different systems on the Web (e.g. a flight information system and a hotel booking system), future interfaces could greatly benefit from these standards.

Thus on the horizon a Web of Services looms – a network of still loosely connected services but with one new and exciting feature: a standardized way of communicating with those services, to send requests and to receive results. This could be the next important step for web technologies – because web services possess many powerful features ideally suited for industrial use and commercial success stories. This also could build the momentum to ensure the wide-spread use of a – in our point of view – very important technology. Current developments support this theory – most new API versions and programming systems supply some sort of Web Services integration (from ancient languages like COBOL to the most recent developments like .NET).

1.3 The Web of Semantics

All afore-mentioned efforts target one underlying and ever present goal: The Semantic Web – an information network of machine-readable data that allows autonomous agencies to gather data, turn it into information, reason about it and come to conclusions. This information network will be traversed by intelligent agents to fuel new and exciting services (see e.g. [24], [25] and [22]). Humans will never be able to fully utilize the mass of data collected in the World Wide Web – thus we need to find new ways to turn all the data into something more than a loosely connected set of HTML pages. The basic building blocks for the semantic web are made up by

- *Semi-structured data*: XML has been accepted as the means of choice to represent platform-independent data in a semi-structured way that allows for an open-ended way of describing data (see [7]). Based on plain text (but powered by Unicode), XML enriches pure data with metadata to allow machines to use the data more effectively and in ways not initially coded into the data format.

- *Machine readable data*: The current proposal for this building block relies on XML as a means of expression and has been named RDF (Resource Description Framework, see [23] and [8]). It should be noted that that RDF has various means of representation but XML seems to be the most natural for the World Wide Web and the most widely used right now. RDF allows to describe resources, properties of resources and relations between resources. RDF can be extended to create more complicated languages and at the same time provides powerful foundations for reasoning (being based on first-order logic). Interestingly RDF takes a very pragmatic approach to provide a viable solution for information representation – it right away allows for inconsistence, incorrectness, and incompleteness in the represented information and takes it as given that data can lead to situations were agents won't be able to come to a decisive or correct conclusion. This pragmatism adheres to the concepts that established the current web – ease of use with an allowance for mistakes.

- *Ontologies* as a means to describe the relations between objects and to define standard hierarchies as descriptions of "the world". A lot of research is concerned with the question of what should be in an ontology language in order to once more find the best way of combining computing and expression power with ease of use. Ontology languages like SHOE (see [17]), DAML (see [20]) and DAML+OIL (see [12]) hint at the power of future metadata structures.

So far the World Wide Web mostly was about encoding data and information. Retrieval, automated reasoning about information, connection of services and basically all other means of exploiting this information pool were only moderately successful. The Web spawned a variety of search engines and meta search engines but these, together with shop systems and web directories, cover the efficient means of access to the World Wide Web for humans. There were some experiments with agents and agent societies (see [X100) but so far these attempts failed to become wide-spread successes due to the lack of a unified information infrastructure and lack of standardized interfaces – CGI (the Common Gateway Interface) is hardly sufficient to build even semi-complex applications in an abstract and elegant way.

To cope with this situation we propose a new model of regarding future applications building on the foundations mentioned so far – a unit of abstraction we have named *Hyperservices*.

2 Hyperservices: A Unified Application Model

When we talk about web-based applications we mean "based on web technologies". Web technologies have been widely accepted and have managed to bring together disparate system structures. We believe that the next important step will be to find a unifying, language- and system-independent architecture that allows for a convergence in current research areas surrounding the Semantic Web.

Looking at the components currently available a unified application model based on agent societies seems to be in reach: The Semantic Web allows to reason about information by structuring information appropriately. This provides the basis for "intelligent" agents (with "intelligence" on a pragmatic hands-on level). Web services introduce the interface for collaboration among systems. Agents are the natural extension to achieve autonomous systems (see [1]). Currently we face a multitude of ontology languages (see e.g. [2]) and many models and theories to map information to efficient data models and retrieval algorithms – but these means will only see wide-spread use if they become easy to communicate to future users (e.g. programmers), based on standard architectures and easy to integrate in existing systems. Integration still is one of the main problems faced by current computer science (from the business perspective) but the Web can only remain successful if it manages to stay commercially interesting (whether by drawing enough people to it to supply Internet Providers with customers or by introducing truly successful E-Business models is not that important). Thus the integration of these new models into existing structures will be the most important task from a business point of view.

Topics under current discussion (e.g. agent societies and the Semantic Web) won't be able to replace classic information systems (e.g. tax accounting, enterprise resource planning and logistics). But if build in the right way they will be able to enrich classic systems by providing added value. They will open up a new venue of information systems – build around the necessity to decide between precision and speed. The sheer size of the Web and its constant flux will make it impossible to store nearly enough data in local systems to allow for efficient information systems (in the classic sense). Thus it seems much more likely that future information systems will be build around the idea of (semi-)autonomous agents wandering across the Web, collecting information, reasoning about it and yielding results – either continuously or after specified resource limits (e.g. time, bandwidth or a financial budget) have been exhausted (see e.g. [19]).

2.1 The WASP Model

We propose a unified model that has been inspired by currently successful component models like e.g. Enterprise JavaBeans and COM. If a way of expressing standard application scenarios can be found that at the same time provides enough structure (to speed up application development) and leaves enough room (to take into account concurrent models, warring philosophies and the general lack of precision in the underlying data in the web environment) an important step has been made towards a truly usable Web of Semantics.

We are convinced that it will not be possible to find the one true way – but e.g. Enterprise JavaBeans have shown a viable approach: start small and target the most pressing issues and then grow to finally encompass most possible scenarios.

Thus our basic philosophy for a unified framework is founded on four building blocks which in our point of view will be absolute necessities to populate the future web with more powerful applications:

<u>W</u>eb Services as a means of providing a unified communication interfaces between applications and agencies (see [32], [9] and [11]).

Agents as a natural and central means to represent typical tasks and solutions for a distributed and constantly changing information environment.

Semantic Web technologies as a means to provide data and information in a consistent manner that allows retrieval and reasoning.

Personalization technologies to customize processes to the needs of the individual user – an absolute necessary concerning the current (and future) size of the World Wide Web lest it becomes impossible to separate useless from useful information.

Agents will be the central building block of this architecture – because they implement the actual business logic. Web Services are the natural means of communication and collaboration for agents working under the described model, the semantic web is the environment (world) for these agents and the personalization rules basically can be used to make up or modify the beliefs of the agents. Thus the described components integrate very nicely and in a very natural manner into the underlying agent paradigm.

The WASP framework will account for a variety of necessities explained in the next sections. In contrast to existing major endeavors in this area (see e.g. [28], [27] and [13] for more details) we plan to provide an architecture that focuses on

- proactive information agents that collect information and provide results by using inference mechanism to reason about the existing information.
- high-level technical support for the application developer (e.g. communication, distribution, data storage),
- tight integration of web technologies (RDF, Web Services, DAML, SOAP, etc.),
- independence from specific kinds of implementations (e.g. no specific communication language will be enforced),
- focus on agents relying on the Semantic Web as the dominant information source.

Thus the following central paradigms will be of greatest importance:

Open Interfaces

Since it is impossible to enforce one true operating system, one true programming language or one true CPU architecture for the network underlying the World Wide Web it is of paramount importance to provide a powerful means of communication between the interacting agencies. SOAP and HTTP (as an underlying protocol) together with Web Services (as a means of interface specification) seem to be natural choices. The framework will provide a layer of abstraction to be able to disconnect from these particular technologies, should e.g. other protocols become more important.

Service Agencies

Agents seem to be a very natural way for describing typical scenarios of web usage. They are the machine representation of human beings who right now have to do most of the work for themselves. Thus the WASP framework will provide means to define a variety of agents – mobile, autonomous, reactive, etc. To enhance the usefulness of the framework it is set up to allow agents to be self-describing – thus automatically turning agents into services that can be used by others and integrated via standard interfaces. This allows for widespread service dissemination and easy integration with other systems.

It will be especially important to integrate current agent research into this framework layer – efforts like DAML (DARPA Agent Markup Language) allow for powerful modeling means to devise agents and agencies.

Data and Information Gathering

The framework must provide for means to accumulate data, compare it and reason about it. Data might be persistent (to allow for agents with increasing reasoning power) or transient (to model short-effect tasks) and data should be interchangeable between different agents. It must be possible to integrate ontologies to allow for a solidified view of the "world" (in regards to the agent or agents).

Personalization Integration

It must be easy to integrate personalization technologies. At the most basic level it should be possible to specify user preferences and dislikes and to integrate them in the reasoning and retrieval process to improve the quality of the returned information.

2.2 The HIVE: Semantic Web Brokering Simplified for WASP Agents

Web servers have been the technical foundation for the success of the World Wide Web. Applications servers have been a successful model in abstracting from the everyday chores of building complex applications and thus form the basis for modern large-scale business applications. Thus it seems natural to evolve to Semantic Information Servers that provide a corresponding environment for semantic web agents specialized on utilizing the Semantic Web resources to provide information services to the end user.

Application servers offer persistence, transactions, distributed processing and scalability if the software complies to a predefined component model (e.g. Java 2 Enterprise Edition / J2EE). This allows developers to focus on the actual task at hand, e.g. implementing the business logic for a complex transaction portal. In our view a similar model is required for semantic web applications based on agent societies. Different layers of abstractions will allow to concentrate on functional requirements and help to abstract from the details of the implementation. In the same way a J2EE application server takes away the details of persistence from the developer, a Semantic Information Server can abstract from the details of e.g. storing semantic information, retrieving it and reasoning about it. This holds true for other areas as well (e.g. information recovery from the web, resource management for agents and communication between members of local and remote agencies). Within the WASP framework we intend to call the Semantic Information Servers a HIVE.

This ideas result in the following infrastructure diagram:

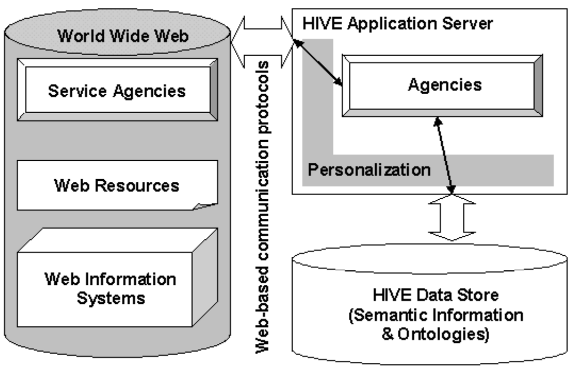

Fig. 1. HIVE Architecture

3 The Current State of Research

In this section we intend to describe but a few of the more important current research topics that need to be solved to further the development of semantic web services:

Ontology integration and translation is a major problem for interconnecting distributed services and systems (see e.g. [18], [16] and [16]): How can differing ontologies for related topics be mapped on each other?

Web Service orchestration, interoperation and transaction handling need to be standardized (see e.g. [33], [34] and [35]).

Standards to allow for personalization need to find wide acceptance (see e.g. [31] and [30] for currently available yet still rarely used standards).

4 Remaining Challenges

Besides the technical questions which currently enjoy most attention a multitude of additional topics needs to be investigated before distributed agent systems and the Semantic Web become truly viable. A few of them are:

- Cost challenges. Who is going to pay for the resources being used in distributed agent networks? It is safe to assume that such agent services will be a lot more cost intensive than the “simple web information platforms of today” (e. g. web servers).
- Pricing challenges. Already now there is a tendency to commercialize high-quality services. How will future information systems be rated in terms of usage fees if the component services of some complex service (e.g. the logistics service, the currency conversion service and the mapping service for a complex online order service) each incur fees but the user of the complex not necessarily will add to the income of the complex service provider (e.g. because the user decides against buying something after getting the shipment information)?
- Business challenges. What are viable business models for Semantic Web agencies and services?
- Quality challenges. How will services be able to guarantee a certain level of quality if they rely on the data collected in the Semantic Web – an information storage that will be as inaccurate as the currently available World Wide Web (mostly because everyone will be able to put up whatever he or she deems correct and appropriate)?
- Trust challenges. How can I be sure that not only the quality of results gained by Semantic Web analysis is sufficient for me but also correct at all?
- Workflow challenges. How can complex workflows (like booking a complete holiday trip) be orchestrated when dynamic service directories, user preferences, potentially faulty information and other factors need to be considered?
- Performance challenges. How must services be constructed to be able to retrieve useful data in a timely manner from the Semantic Web – a web that is infinitely more complex to search compared to current

search engines and technologies due to the far more involved complexity created by allowing inferences, conclusions and reasoning about information?
- Security challenges. How can personal and private information be protected from prying eyes?
- Legal challenges. Who will be held responsible for incorrect, imprecise or faulty information derived from or modified by Semantic Web content?
- Architectural challenges. What is the best software infrastructure / application architecture to allow for a rapid dissemination of the technologies involved? How can user and developer acceptance be increased?

5 Conclusion

As the short list in the previous section demonstrates there are monumental challenges lying ahead. We intend to take the next small step by providing a technical "best of breed" infrastructure for developing, testing and quickly distributing new ideas in the areas of web services, agent technologies, Semantic Web and personalization. As explained such architecture should use the same virtues that made the World Wide Web a success: it must be easy to understand, open for future development philosophies, quick to provide results and powerful enough to be useful for a developer by providing substantial base functionality and allowing to concentrate on the actual business problems.

References

[1] Benjamins, Richard: Agents and the Semantic Web: A Business Perspective. Agent Technology Conference (ATC); 2002.

[2] Benjamins, V. R., Fensel, D., Asunción G. P.: Knowledge Management through Ontologies

[3] Berners-Lee, Tim; Hendler, James; Lassila, Ora: The Semantic Web, Scientific American, Mai 2001.

[4] Berners-Lee, Tim: Semantic Web Roadmap, Online available at http://www.w3c.org/DesignIssues/Semantic.html, September 1998.

[5] Berners-Lee, Tim: Weaving the Web, TEXERE Publishing Limited, London, 2000.

[6] Bray, T.; Paoli, J.; Maler, Eve; Sperberg-McQueen, C. M., Paoli, Jean (Editoren): Extensible Markup Language (XML) 1.0, W3C Recommendation, available online at http://www.w3c.org/TR/REC-xml, (6. Oct. 2000).

[7] Bray, T.; Paoli, J.; Maler, Eve; Sperberg-McQueen, C. M., Paoli, Jean (Editors): Extensible Markup Language (XML) 1.0, W3C Recommendation (6. Oct. 2000), available online at http://www.w3c.org/TR/REC-xml, Februar 1998.

[8] Brickley, Dan; Guha, R. V. (Edt.): Resource Description Framework (RDF) Schema Specification 1.0, W3C Candidate Recommendation (27. March 2000), available online at http://www.w3c.org/TR/RDF-schema, World Wide Web Consortium, March 2000.

[9] Christensen, Erik; Curbera, Francisco; Meredith, Greg; Weerawarana, Sanijva: Web Services Description Language (WSDL) 1.1, W3C Note (15. March 2001), Online available at http://www.w3c.org/TR/wsdl, World Wide Web Consortium, March 2001.

[10] Ciancarini, P., Wooldridge M. J. (editors): First International Workshop on Agent-Oriented Software Engineering, volume 1957 of Lecture Notes in Computer Science. Springer, Berlin, 2001.

[11] Dale, Jonathan: Exposing Web Services, Agentcities.NET iD2, Agentcities Lisbon, 9 - 10 September, 2002.

[12] DAML+OIL. http://www.daml.org/language/, 2000.

[13] Finin, T., Labrou, Y., Mayfield, J.: KQML as an agent communication language. In Software Agents, Bradshaw JM (ed.), MIT Press, pp.291-316, 1997.

[14] Gottschalk, K., Graham, S., Kreger, H., Snell, J.: Introduction to Web services architecture, IBM Systems Journal, Volume 41, Number 2, 2002.

[15] Gourley, D., Totty, B.: HTTP: The Definitive Guide, O'Reilly, 2002.

[16] Gruber, T. R.: A translation approach to portable ontologies. Knowledge Acquisition, 5(2):199-220, 1993.

[17] Heflin, J., Hendler, J., and Luke, S.: SHOE: A Knowledge Representation Language for Internet Applications. Technical Report CS-TR-4078 (UMIACS TR-99-71), Dept. of Computer Science, University of Maryland at College Park. 1999.

[18] Heflin, J.; Hendler, J.: Dynamic Ontologies on the Web, Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000). AAAI/MIT Press, Menlo Park, CA, 2000.

[19] Hendler, J.: Agents and the Semantic Web, IEEE Intelligent Systems, Vol. 16, Issue 2, März / April 2001.

[20] Hendler, J.; McGuiness, Deborah L.: The DARPA Agent Markup Language, IEEE Intelligent Systems, Vol. 15, Issue 6, November / December 2000.

[21] Huhns, Michael N.; Stephens, Larry M.: Consensus Ontologies – Reconciling the Semantics of Web Pages and Agents, IEEE Internet Computing, Vol. 5, Issue 5, September / October 2001.

[22] Joy, Bill: Shift from Protocols to Agents, IEEE Internet Computing, Vol. 4, Issue 1, January / February 2000.s: 2440, The Internet Society, 1998, http://www.ietf.org/rfc/rfc2440.txt

[23] Lassila, Ora: The Ressource Description Framework, IEEE Intelligent Systems, Vol. 15, Issue 6, November / Dezember 2000.

[24] McIlraith, Sheila A.; Son, Tra Cao; Zeng, Honglei: Semantic Web Services, IEEE Intelligent Systems, Vol. 16, Issue 2, March / April 2001.

[25] Metcalfe, Bob: The Next-Generation Internet, IEEE Internet Computing, Vol. 4, Issue 1, January / February 2000.

[26] Mitra, Nilo (Editor): SOAP Version 1.2, Part 0: Primer, Part 1: Messaging Framework, Part 2: Adjuncts, W3C Working Draft (18. Dec. 2001), online available at http://www.w3c.org/TR/soap-part0, http://www.w3c.org/TR/soap-part1, http://www.w3c.org/TR/soap-part2, World Wide Web Consortium, 2001.

[27] Object Management Group, http://www.omg.org

[28] O'Brien, P. D., Nicol, R., C.: FIPA - towards a standard for software agents. BT Technology Journal, Vol.16, no.3, pp. 51, July, 1998

[29] Parunak, H. V. D.: Practical and Industrial Applications of Agent-Based Systems, Industrial Technology Institute, 1998.

[30] Platform for Internet Content Selection (PICS), http://www.w3.org/PICS

[31] The Platform for Privacy Preferences 1.0 (P3P1.0) Specification, W3C Recommendation 16 April 2002, http://www.w3.org/TR/2002/REC-P3P-20020416/

[32] UDDI Technical White Paper, Online verfügbar unter http://www.uddi.org, UDDI.org, 2001.

[33] Web Services Choreography Requirements 1.0, W3C Working Draft 12 August 2003, http://www.w3.org/TR/ws-chor-reqs

[34] Web Services Interoperability Standard WS-I Basic Profile 1.0, http://www.ws-i.org

[35] Web Services Transaction (WS-Transaction), 9. August 2002, http://www-106.ibm.com/developerworks/library/ws-transpec/