Intelligent "Mirror Web Browsing" vs. Pull/Push Technology

The growth of overall Web content size results to multiple problems in managing various Web usage scenarios. Any initiator of a business transaction in the Web, e.g. a customer in "pull"-type scenarios or a merchant in the "push"-type scenarios, have to spend more time online nowadays than it is reasonable. Emerging Semantic Web and Intelligent Agents technologies are the good starting point to simplify Web content management by switching online presence focus from humans to machines. In this paper we consider possible classification of Web content retrieval architectures based on features provided by availability of agents and Semantic Web based content annotations. We have shown that there is specific family of so-called "mirror" architectures and appropriate scenarios, where the initiator of any transactions is neither merchant nor customer but a mediator (active information broker agent). Such architectures are shown to be the most appropriate ones for the sake of removing humans from online presence while retrieving Web content. Business interpretation of mirror scenarios is provided.

Vagan Terziyan
Department of Mathematical Information Technology University of Jyvaskyla
P.O. Box 35 (Agora),
40014 Jyvaskyla
FINLAND
E-mail: vagan@it.jyu.fi

1. Introduction

Our information world is becoming more connected, different type of devices are used as entry points into the WWW, Internet usage is growing explosively and mobile data services are reaching users worldwide. When people first began to surf the Internet using their web browsers, they used what is known as "pull technology". If you want information from the web, you need to get online with a web address and point your browser to that address. In this fashion, you are "pulling" the information by yourself. The crucial problem however is an exponential growth of the amount of online available information during the recent years, and even not the growth itself but the fact that the amount of data has reached certain threshold when most of ordinary methods and tools used for information management have became useless. The WWW currently contains about 3 billion static documents, which are accessed by over 300 million users [2]. The first victims of this vast amount of information are the users of it, for example:

- a user nowadays waits longer online during search of information from submitting search query until getting search results;
- the probability that search result contain information, a user is looking for, decreases and thus, to be sure that enough appropriate sites were found, a user should spend online more time trying different search queries;
- the amount of returned search results increases and thus the time a user should spend online selecting needed information also increases;
- the amount of appropriate sites of an average search in the web increases as well as the time a user should spend online to download and integrate selected information to one document for further use.

If we do not want to create a new generation of "information slaves" from humans who might spend tens of years in the Internet online, then some radical changes to Internet technologies should be performed. Among the expectations for revolutionary improvements of the situation are collaborative efforts towards the Semantic Web with appropriate agent technologies, web services, information integration, personalization and filtering tools, etc. The problems and urgent needs attract also businesses to the web to interconnect appropriate customers, manage and integrate web resources and services and benefit from this. E-Commerce is one of the driving forces for future commerce development including also public commerce [1], i.e. the case when Internet users are involved in the e-commerce activity.

However still context of the problem remains the same: a web user today (one of about $3 \cdot 10^8$) is looking for a web resource (i.e. spending time and money) among about $3 \cdot 10^9$ different resources, whatever he uses as support tools for that. It is also most probable that the amount of the online resources and services in the web will grow much faster than the amount of their users. Our question is: "is that so that a user is more interested to find and utilize a resource than this resource (i.e. people behind) "is interested" to be used and utilized by a user?" If answer to this question is negative, then the above context is not appropriate. The new context would be that a web resource itself is looking (i.e. spending time and money) for users that are interested in it.

One of the most promising technologies today on the Internet, which try to fit this new context, would be the emerging "push" technology, which introduces a completely new model to information distribution/retrieval applications on the Internet. Push Technology is different from pull technology as it enables to push information out to people interested in ones information, products or services. It's really a specialized way of reaching people who have requested or are members of special interest groups and having information (or service) advertisement reach them. With push technology a server sends information to a client without waiting its retrieve message. User after preliminary subscription for necessary information will receive it in real time. Perspectives of this technology look promising. First, a user doesn't need to spend time and money on finding the information. Secondly, the authorization problem is solved automatically [5].

To facilitate the process of information retrieval the information brokers [10] are used as intermediaries between users and sources. An information broker takes input from information providers as well as information consumers (in the form of advertisements and queries), it then may enrich this input with additional (meta) information and will try to best match the input with the most fit parties for it [11]. Broker agents are assumed to communicate and negotiate when necessary with user agents and source agents since agent technology fits well in the information discovery paradigm (Figure 1a). The connection between electronic commerce and information retrieval via information brokers is discussed in [12]. The role of so-called middle-agents - information brokers is shown to describe the concept of mediated information market. Figure 1b shows three research areas combined in agent-based information discovery according to [12]. They are Information Discovery, Middle-agents, and Electronic Commerce. Region 1 on the picture represents a client-server information market, region 2 corresponds to mediated market, region 3 stands for mediated information discovery, and region 4 expresses a mediated information market.

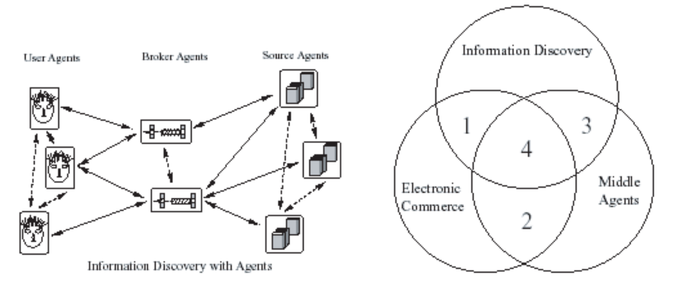


Figure 1. (a) Broker agents paradigm, (b) related research areas [12]

In [7] authors discussed problems related to online information search in the Web and also concluded in favour of the use of information agents. They present an overview of intelligent software agents in information retrieval, including an explanation of agents and agent architectures, and present several agent systems. Authors concluded that the future of information agents in database information retrieval and in Web search is promising, because there are too many sources for a person sitting at a computer and searching specific information and that it is much easier to submit a query to an agent and let it find the information. This supposed to save both time and frustration in following links across the Internet, and it promises a "bright future for intelligent search agents" [7]. However such optimism recently is being threaten by continuously growing and heterogeneous Web content, which is too difficult as such for agents to be properly selected, processed and integrated. This is simple - Web was initially meant for humans and not for agents. Web should be adapted to the agents if we want to use them effectively.

The Semantic Web is an initiative of the World Wide Web Consortium (W3C), with the goal of extending the current Web to facilitate a universally accessible content. Current trends on Web development leading to a more sophisticated architecture: Semantic Web; Device Independence; Web Services. Tim Berners-Lee [6] has a vision of a semantic web, which has machine-understandable semantics of information, and trillions of specialised reasoning services that provide support in automated task achievement based on the accessible information. Semantic Web is the vision of having data on the Web defined and linked in a way that it can be used by machines not just for display purposes, but for automation, integration and reuse of data across various applications. This vision assumes annotating Web resources with machine-interpretable descriptions of their underlying semantics, and provides mechanisms for automated reasoning about them. This gives a completely new perspective for the content retrieval in the Web, which enables automation of the content use by information agents. In the same time availability of semantic annotations also for the Web users' preferences allows providing quality Web content personalization. On the technology side, Web-enabled languages and technologies are being developed (e.g. RDF, RDF-Schema, DAML+OIL, OWL, DAML-S and others), schema and ontology integration techniques are being examined and refined.

In this paper we consider possible classification of Web content retrieval architectures based on features provided by availability of agents and Semantic Web based content annotations (Chapter 1 and appendixes). We have shown that there is specific family of so-called "mirror" architectures and appropriate scenarios (e.g. Public Commerce - Chapter 3 and Intelligent Mirror Browsing - Chapter 4), where the initiator of any transactions is neither merchant nor customer but a mediator (active information broker agent). Such architectures are shown to be the most appropriate ones for the sake of removing humans from online presence while retrieving content.

Business interpretation of mirror scenarios is provided (Chapter 5). Conclusions are in Chapter 6.

2. General Classification of Information Retrieval Architectures

In this chapter we are providing general classification of different information retrieval architectures to be able to discover different e-commerce scenarios associated with these architectures.

We use the following attributes and their values for the classification:

- Attribute 1: Availability and location of semantic annotations for users (preferences, profiles, search queries, etc.). Values: (1) Absent; (2) Server-Sided; (3) Client-Sided;
- Attribute 2: Availability and location of semantic annotations for Web resources (annotations, descriptions, advertisements, etc.). Values: (1) Absent; (2) Server-Sided; (3) Client-Sided;
- Attribute 3: Availability and location of intelligent Web browsing/search/integration agent (application, service). Values: (1) Absent; (2) Server-Sided; (3) Client-Sided;
- Attribute 4: The basic type of the technology used for information retrieval. Values: Pull; (2) Push; (3) "Mirror". The last "mirror" value means intelligent integration of pull and push technologies.

In Appendix A you can see all 81 (3x3x3x3=81) possible combination of attribute features and appropriate architectures. From that set we have selected and marked 21 valid/reasonable/principally different architectures. The selection of these 21 valid architectures was based on the following restrictions applied to the appropriate attributes values.

Restrictions on "pull" architectures:

- Assume that if semantic annotations of the resources are available then there should be an agent (either in user client site or in external service site) to explore these annotations;
- Assume that if resources are not annotated then there is no need to involve agents to the architecture;
- Assume that there is no need to have annotation of user preferences for such architectures.

Restrictions on "push" architectures:

- Assume that if semantic annotations of the users preferences are available then there should be an agent (either in the resource client site or in external service site) to explore these annotations;
- Assume that if user preferences are not annotated then there is no need to involve agents to the architecture;
- Assume that there is no need to have annotation of the resources for such architectures.

Restrictions on "mirror" architectures:

- Assume availability of a mediation service with a service agent;
- Assume availability of either semantic annotation of a user preferences or semantic annotation of a resource content;
- Assume that the service is the only initiating point of all transactions (this e.g. means that there are no direct transactions between a user and a resource).

The pictures, which generally illustrate the 21 selected architectures, are presented in Appendix B.

We will provide more details towards description of two "mirror" type architectures (number 45 - "Public Commerce" and number 81 "Intelligent Mirror Browsing") and discuss their possibilities for the e-commerce applications.

3. Public Commerce Architecture

According to the above classification Public Commerce architecture assumes availability of semantic annotations of users preferences and semantic annotations of Web resources or services. It also assumes that these semantic annotations are submitted to and stored in some P-Commerce service server. Also it is supposed that a matchmaking agent of the P-Commerce service will find needed matches, integrates and delivers data and services, manages appropriate transactions (See Figure 2).

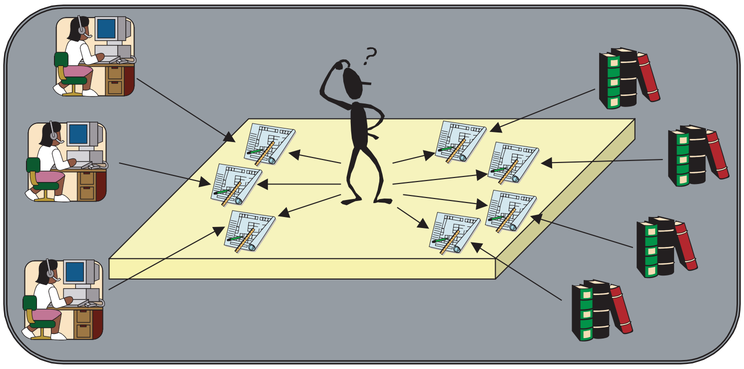


Figure 2. "P-Commerce": general architecture

Possible example scenario for the transactions within P-Commerce architecture is presented in Figure 3.

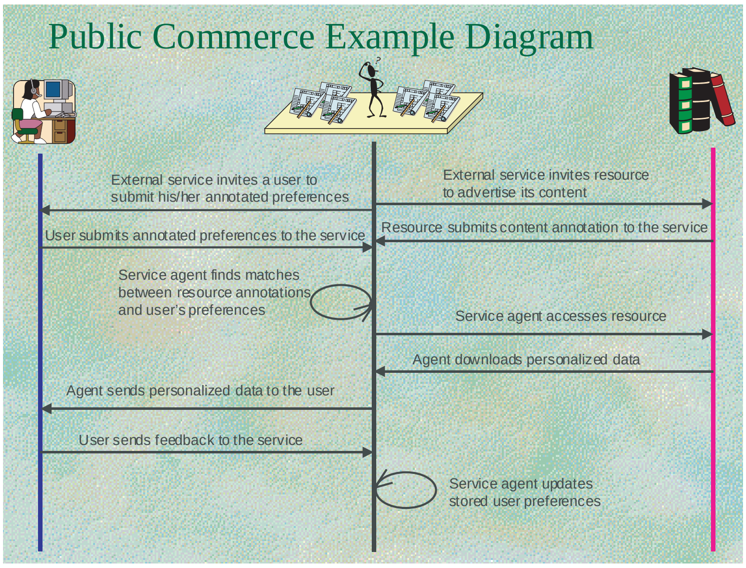


Figure 3. P-Commerce: typical scenario

The e-commerce implementation of the P-Commerce approach [1] is based on the assumption that every person in the society in some way more or less participates in public business process. "We are buying something from another persons watching newspapers with public offer-

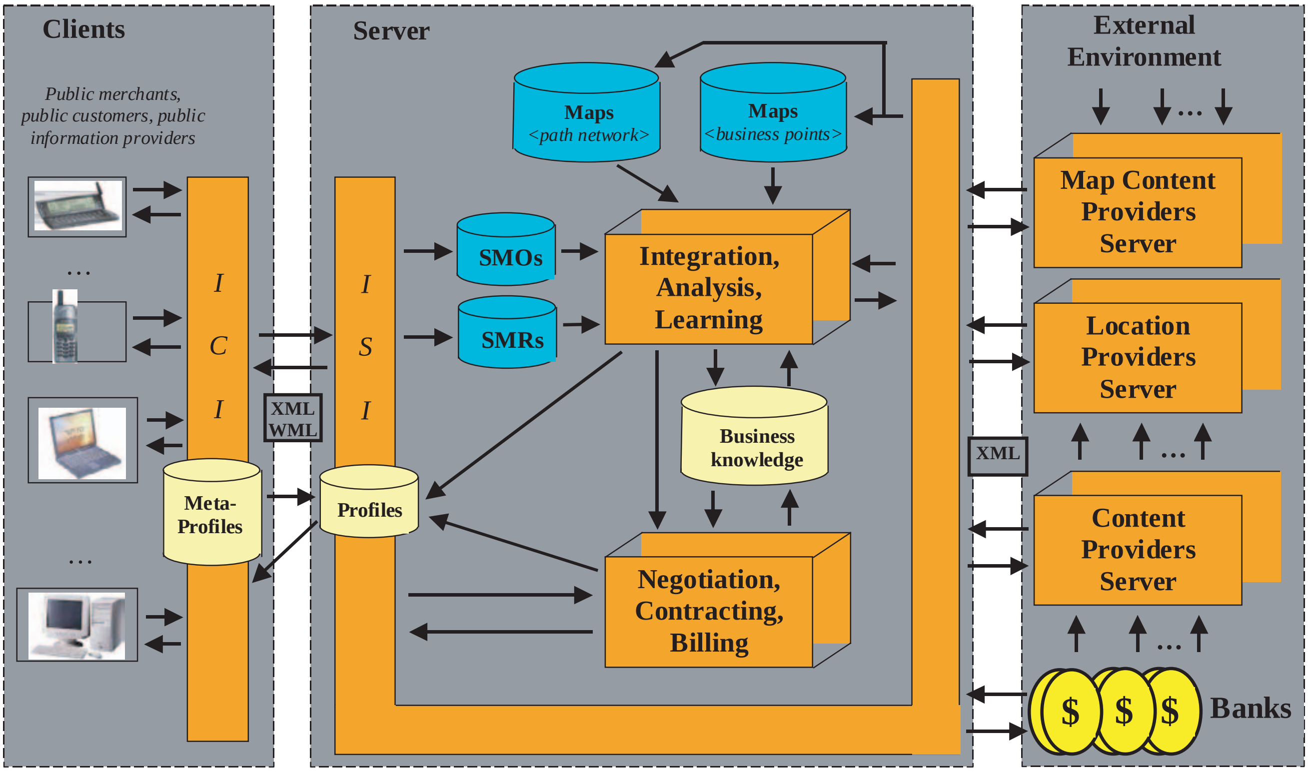


Figure 4. Possible business implementation of P-Commerce architecture according to [1]

ings or we are selling something unnecessary making announcements in numerous places by numerous ways. Some of us even sometimes make small business by finding in news such offerings and requests that fit each other". Essential role in p-commerce activities plays also the location information, which can be available e.g. for mobile users. Two types of clients in P-commerce were considered: business clients (public merchants, information providers) and customers clients (users). It was supposed that there is restricted number of business profiles in p-commerce. The p-commerce application should help its business clients to find out their profiles from open profile database. Profile is certain frame by filling which one can specify his offering (request) for p-commerce. It is assumed that mobile business client represents interests of some company or some private interest and making appropriate short mobile offerings (SMO). Mobile business client software supports a user to classify his SMO as precise as possible, downloads profile (frame) from application server appropriate to the class and taking into account mobile device type, translates profile filled (interpreted) by the user to XML/RDF format and submits it the P-commerce application server. A mobile customer client can also represent interests of some company or his own private interests and making appropriate short mobile requests (SMR) about his needs. Mobile business client software supports a user to classify and submit his SMR in a similar way as in the case of SMO. Client part of the architecture consists on intelligent client interface (ICI) software and meta-profiles knowledge base. ICI should provide possibility to adapt incoming data to certain mobile device interface, to represent geographical data in the form of maps, to provide brief description of objects within some neighborhood around the user and show the way to reach them, to select, download and visualize profiles from server, to support semantic annotations of offering and request data. Meta-profiles knowledge base includes profiles of the highest layer of their classification, which being interpreted are necessary to select profiles of the lower classification layers from the server. Server part of the architecture consists on intelligent server interface (ISI) software, SMOs and SMRs databases, maps content databases, business knowledge base, profiles knowledge base, processor for data integration, analysis, and learning (data mining and knowledge discovery), and processor for negotiations, contracting and billing. ISI controls external transactions during profiling of clients, negotiations, contracting and billing between clients, converts formats of external inputs to internal format, completes SMOs and SMRs databases, supplies clients with requested profiles from the profiles knowledge base. SMRs is an open database, which collects requests of mobile clients. Objects of this database are active in a sense that for every incoming request data analysis processor tries to satisfy it (i.e. to find out nearest neighbor offering from SMOs database) and if this attempt fails then this object will be kept active until something appropriate comes to SMOs database (see Figure 4).

4. Intelligent Mirror Browsing Architecture

According to the above classification, "Intelligent Mirror Browsing" looks very similar to the Public Commerce architecture. However there are some major differences. Intelligent Mirror Browsing architecture assumes that semantic annotations either of resources or user preferences are stored locally (at a client side). Also it is supposed that a matchmaking agent of the P-Commerce service will himself browse and find these annotations, discovers ones, which match each other, downloads, integrates and delivers personalized data and services to users (See Figure 5).

Possible example scenario for the transactions within such architecture is presented in Figure 6.

As one can see such abstract architecture makes a user free from online web browsing or querying as well as from

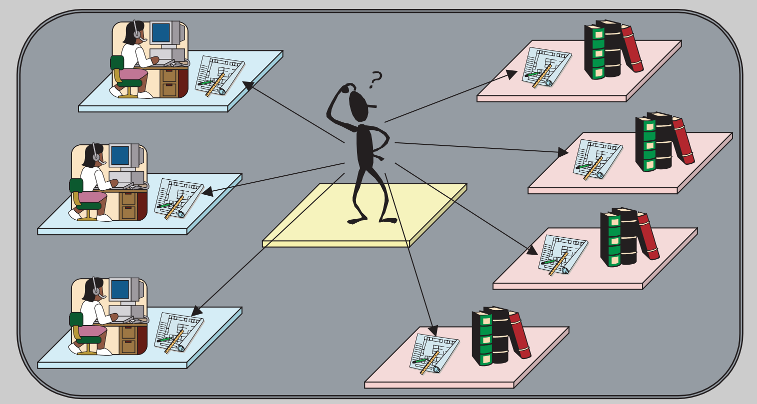


Figure 5. "Intelligent Mirror Browsing": general architecture

unnecessary losses of resources to do this. Moreover, even web information or service providers should not waste resources and target their customers or advertise themselves whenever it possible. So you just fill your request or advertisement and do not even think where to submit it. Keep it locally and be sure that Intelligent Web Browsing service will find it, make all online jobs by finding, downloading, integrating, personalizing and delivering appropriate data and services. Also it looks nice to keep your personal data locally with easier to update and easier to protect privacy possibilities.

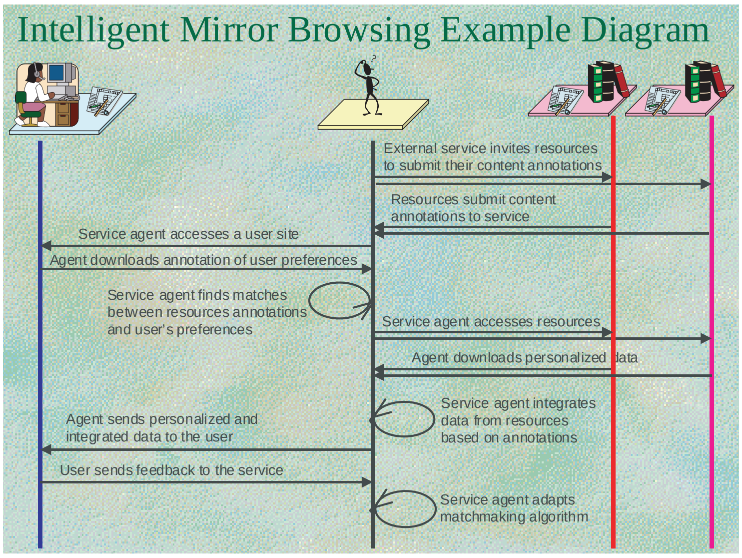


Figure 6. Intelligent Mirror Browsing: typical scenario

However there are some questions to be raised. What might be the business reason to anyone to launch that type of services, who and how will pay for this? What kind of available technologies should be implemented to make it work? What might happen if there would be plenty of such services, which will try to deliver to a user simultaneously something he has requested? These problems we will try to address in the following chapters.

5. Business Motivation for Mirror Architectures

Commerce simply is the exchange of money for goods or services between companies and end consumers. Therefore, "e-commerce" is doing commerce using electronic technology, e.g. Internet [3]. The basic idea is quite simple: one finds what is needed in the Internet, requests this from appropriate service provider, gets and pays. Anyway that e-commerce player who proposes something via Web is intended to raise Internet users' interest towards services or goods he proposes. Advertisement is the main among known instruments to attract attention of the particular users. Push and pull models of advertisement and information delivery make it possible to distribute information on the Web to a vast amount of customers. Finally of course it often happens that the service or goods that are delivered to a particular customer are not as satisfactory as they were advertised.

Now, lets consider a very common situation when a usual person is willing to sell or buy something personal. Placing an announcement in appropriate newspapers is one of oldest and still popular possibilities. The success of such business operation either selling or buying depends on the assumption that an appropriate person will read the announcement. What if wanted person will not read your announcement? This case allows appearance for another commerce entity. This entity is a third party person who reads all the available announcements and finds best matching pairs of them. After that, he informs both parties of the matching and get a percentage from the deal. He is not exactly a mediator or a middleman in usual sense. This is because the last ones are usually recruited (by either a consumer or a merchant) to do some mediation job and they discuss their estimated profit in advance. However in our case a "mediator" initiates business transactions with assumed consumer and merchant himself and takes a risk not to be grateful for this job from either party. Such behaviour fits well the "mirror" type of architectures presented in previous chapter. Figure 7, represents this kind of business process with a "mirror mediator" instance.

Internet provides have more powerful facilities to do such kind of mirror mediation business in the cyberspace. There are Web services that allow users to publish the ads and e.g. themselves play a role of mediators by performing announcements matching. Available services of this type (see e.g. [1]) require both users - consumers and users-merchants to register and publish an announcement.

Such mirror-mediation instance ("active broker") might be a good interpretation of an e-market place concept, e.g. [4], and it is providing new smarter collaborative environment, which takes ini-

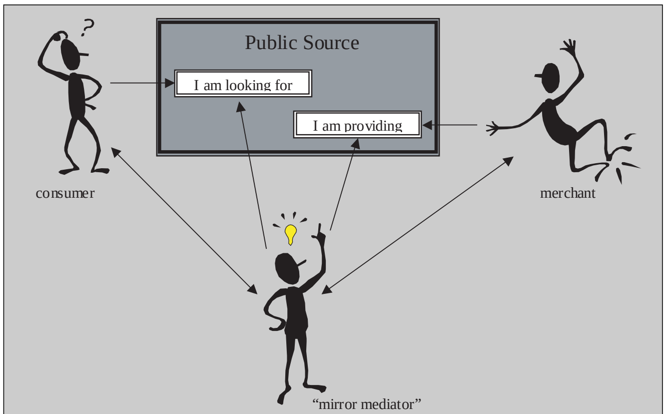


Figure 7. Simple "Mirror Middleman" instance

tiative and semantically links together multiple business parties.

Automatic electronic markets are helpful if the match between providers and requesters must be made fast and/or there is a large volume of transactions. In e.g. tourism information systems both criteria apply. It is shown in [8] how the vision of the Semantic Web and already existing Semantic Web technologies can be used for next-generation tourism information systems. This requires a rich conceptual model about the tourism domain such that the benefits of electronic markets may be applied in tourism. The discussion is provided about how the final customers let their agents trade against the final providers with agencies providing the market place and the integrated information.

One of the main challenges for e-commerce infrastructure designers is not only retrieving data from different Web sources but also obtaining an integrated view that can overcome any contradictions or redundancies. Example of appropriate mediation instance, so-called virtual catalog, for that case is presented in [9]. Virtual catalogs act as instruments to retrieve information dynamically from multiple catalogs and present unified product data to customers. Instead of having to interact with multiple heterogeneous catalogs, customers can instead interact with a virtual catalog in a straightforward, uniform manner. In this case such architecture will be similar to the Architecture # 16: "Semantic Search from Server" (See Appendix B). However, if the virtual catalog is self-creative active instance, which is updates itself based on multiple catalogs browsing and in the same time takes initiative to interact with users' annotations, then such instance can be also classified as one of mirror schemes (See Architecture 72 "Smart Mirror" in Appendix B).

6. Conclusions

In this paper we try to classify and compare various scenarios of managing distributed Web resources, i.e. retrieving, filtering, integration and delivery of information from Web resources. We select three basic features for the classification. First one is based on the application of agent technology and we restrict consideration on the availability either user agent, source agent or mediator agent (information broker). The second feature is based on application of the Semantic Web technology and we consider availability of either semantic annotation of Web resources, semantic annotation of users (e.g. profiles, preferences, etc.) or both. The last feature is based on transaction management paradigm and it varies depending on who is initiating content management transaction: user, source or mediator. The classification has produced 81 scenarios, from which 21 scenarios were selected as principally different. Then we focused our consideration to those scenarios, which has fixed third feature, i.e. when the initiator of transaction is a mediator. We call this group of scenarios and appropriate architectures as "mirror" architectures and we specifically presented two most advanced of them: Public Commerce (with centralized semantic annotations) and Intelligent Mirror Browsing (with decentralized annotations). We have shown that these architectures are principally different from known pull, push, and information broker architectures just because the transactional initiative of a mediator provides new and quite important possibilities. We tried to provide business interpretation for mirror architectures. We believe that such architectures can simplify Web content management by switching online presence focus from humans to machines and from main e-commerce players to mediators.

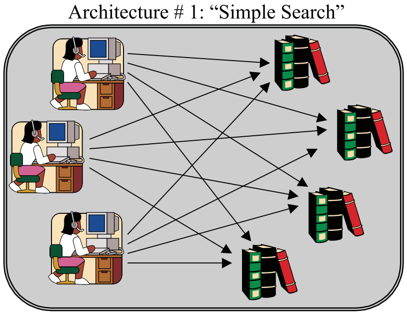
Appendix A: Classification of Information Retrieval Architectures

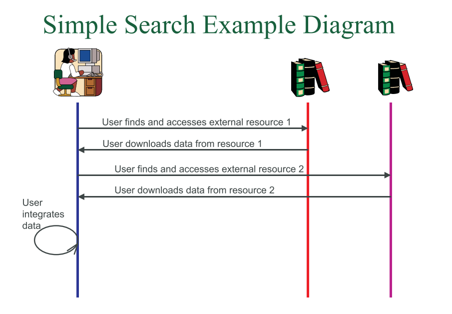
#	User Annotation	Resource Annotation	Browsing / Search Agent	Technology	Comment
1	**Absent**	**Absent**	**Absent**	**Pull**	**Simple search**
2	Absent	Absent	Absent	Push	
3	Absent	Absent	Absent	Mirror	
4	**Absent**	**Absent**	**Client-Sided**	**Pull**	**User agent based search**
5	**Absent**	**Absent**	**Client-Sided**	**Push**	**Resource agent based push**
6	Absent	Absent	Client-Sided	Mirror	
7	**Absent**	**Absent**	**Server-Sided**	**Pull**	**Service agent based search**
8	**Absent**	**Absent**	**Server-Sided**	**Push**	**Service agent based push**
9	Absent	Absent	Server-Sided	Mirror	
10	Absent	Server-Sided	Absent	Pull	
11	Absent	Server-Sided	Absent	Push	
12	Absent	Server-Sided	Absent	Mirror	
13	**Absent**	**Server-Sided**	**Client-Sided**	**Pull**	**Semantic search from client**
14	Absent	Server-Sided	Client-Sided	Push	
15	Absent	Server-Sided	Client-Sided	Mirror	
16	**Absent**	**Server-Sided**	**Server-Sided**	**Pull**	**Semantic search from server**
17	Absent	Server-Sided	Server-Sided	Push	
18	**Absent**	**Server-Sided**	**Server-Sided**	**Mirror**	**Blind centralized mirror**
19	Absent	Client-Sided	Absent	Pull	
20	Absent	Client-Sided	Absent	Push	
21	Absent	Client-Sided	Absent	Mirror	
22	**Absent**	**Client-Sided**	**Client-Sided**	**Pull**	**Smart search from client**
23	Absent	Client-Sided	Client-Sided	Push	

24	Absent	Client-Sided	Client-Sided	Mirror	
25	**Absent**	**Client-Sided**	**Server-Sided**	**Pull**	**Smart search from server**
26	Absent	Client-Sided	Server-Sided	Push	
27	**Absent**	**Client-Sided**	**Server-Sided**	**Mirror**	**Blind distributed mirror**
28	Server-Sided	Absent	Absent	Pull	
29	Server-Sided	Absent	Absent	Push	
30	Server-Sided	Absent	Absent	Mirror	
31	Server-Sided	Absent	Client-Sided	Pull	
32	**Server-Sided**	**Absent**	**Client-Sided**	**Push**	**Semantic push from client**
33	Server-Sided	Absent	Client-Sided	Mirror	
34	Server-Sided	Absent	Server-Sided	Pull	
35	**Server-Sided**	**Absent**	**Server-Sided**	**Push**	**Semantic push from server**
36	**Server-Sided**	**Absent**	**Server-Sided**	**Mirror**	**Simple mirror**
37	Server-Sided	Server-Sided	Absent	Pull	
38	Server-Sided	Server-Sided	Absent	Push	
39	Server-Sided	Server-Sided	Absent	Mirror	
40	Server-Sided	Server-Sided	Client-Sided	Pull	
41	Server-Sided	Server-Sided	Client-Sided	Push	
42	Server-Sided	Server-Sided	Client-Sided	Mirror	
43	Server-Sided	Server-Sided	Server-Sided	Pull	
44	Server-Sided	Server-Sided	Server-Sided	Push	
45	**Server-Sided**	**Server-Sided**	**Server-Sided**	**Mirror**	**Public commerce**
46	Server-Sided	Client-Sided	Absent	Pull	
47	Server-Sided	Client-Sided	Absent	Push	
48	Server-Sided	Client-Sided	Absent	Mirror	
49	Server-Sided	Client-Sided	Client-Sided	Pull	
50	Server-Sided	Client-Sided	Client-Sided	Push	
51	Server-Sided	Client-Sided	Client-Sided	Mirror	
52	Server-Sided	Client-Sided	Server-Sided	Pull	
53	Server-Sided	Client-Sided	Server-Sided	Push	
54	**Server-Sided**	**Client-Sided**	**Server-Sided**	**Mirror**	**Semantic Web mirror**
55	Client-Sided	Absent	Absent	Pull	
56	Client-Sided	Absent	Absent	Push	
57	Client-Sided	Absent	Absent	Mirror	
58	Client-Sided	Absent	Client-Sided	Pull	
59	**Client-Sided**	**Absent**	**Client-Sided**	**Push**	**Smart push from client**
60	Client-Sided	Absent	Client-Sided	Mirror	
61	Client-Sided	Absent	Server-Sided	Pull	
62	**Client-Sided**	**Absent**	**Server-Sided**	**Push**	**Smart push from server**
63	**Client-Sided**	**Absent**	**Server-Sided**	**Mirror**	**Preference-based mirror**
64	Client-Sided	Server-Sided	Absent	Pull	
65	Client-Sided	Server-Sided	Absent	Push	
66	Client-Sided	Server-Sided	Absent	Mirror	
67	Client-Sided	Server-Sided	Client-Sided	Pull	
68	Client-Sided	Server-Sided	Client-Sided	Push	
69	Client-Sided	Server-Sided	Client-Sided	Mirror	
70	Client-Sided	Server-Sided	Server-Sided	Pull	
71	Client-Sided	Server-Sided	Server-Sided	Push	
72	**Client-Sided**	**Server-Sided**	**Server-Sided**	**Mirror**	**Smart mirror**
73	Client-Sided	Client-Sided	Absent	Pull	
74	Client-Sided	Client-Sided	Absent	Push	
75	Client-Sided	Client-Sided	Absent	Mirror	
76	Client-Sided	Client-Sided	Client-Sided	Pull	
77	Client-Sided	Client-Sided	Client-Sided	Push	
78	Client-Sided	Client-Sided	Client-Sided	Mirror	
79	Client-Sided	Client-Sided	Server-Sided	Pull	
80	Client-Sided	Client-Sided	Server-Sided	Push	
81	**Client-Sided**	**Client-Sided**	**Server-Sided**	**Mirror**	**Intelligent mirror browsing**

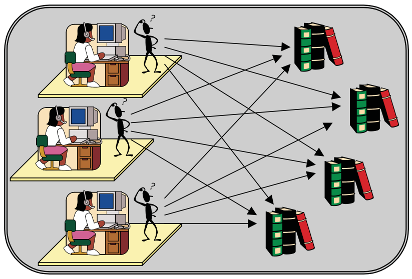
Appendix B: Pictures of 21 Information Retrieval Architectures and Sample Transactions

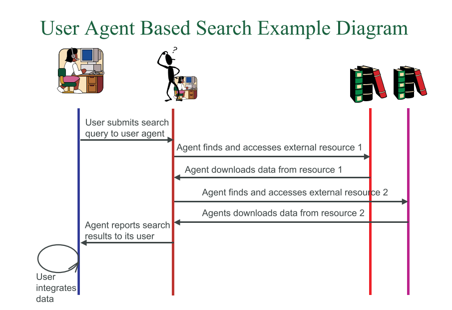
Architecture # 1: "Simple Search"



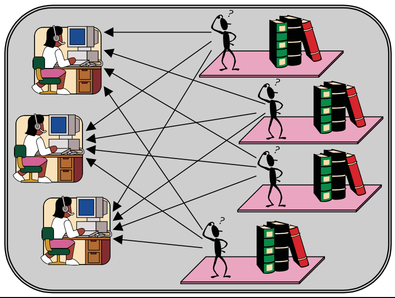


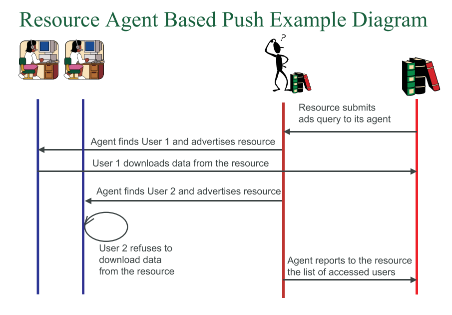
Architecture # 4: "User Agent Based Search"



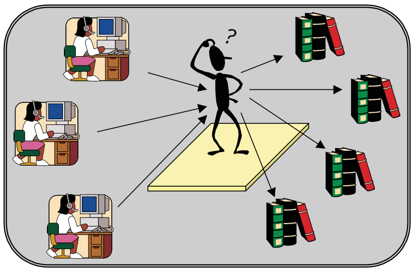


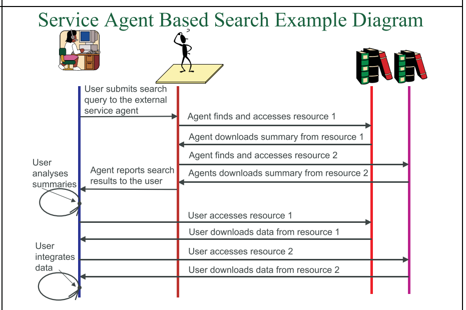
Architecture # 5: "Resource Agent Based Push"



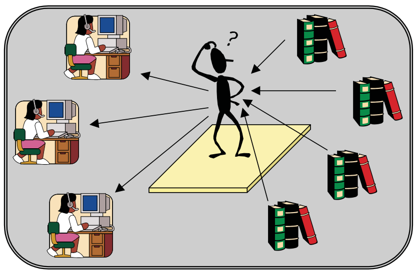


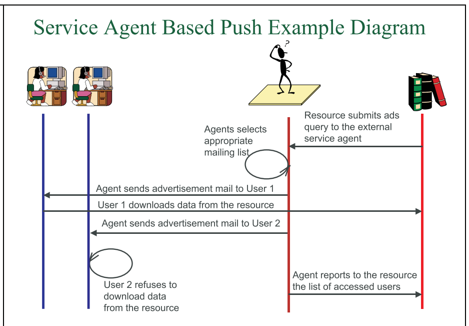
Architecture # 7: "Service Agent Based Search"



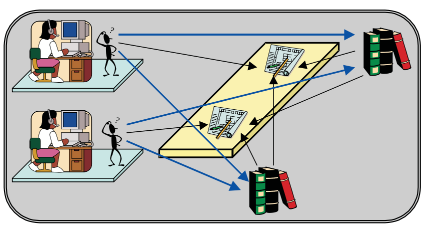


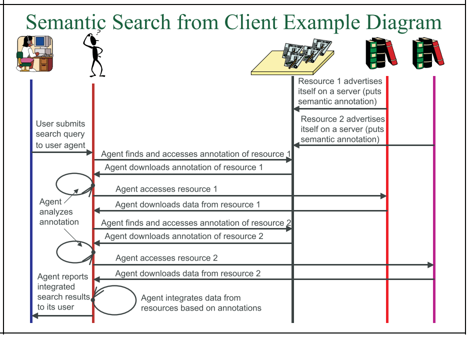
Architecture # 8: "Service Agent Based Push"



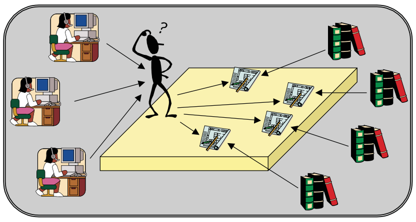


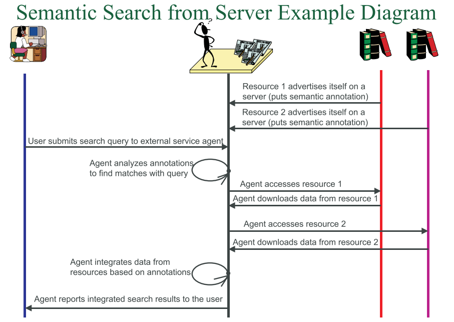
Architecture # 13: "Semantic Search from Client"



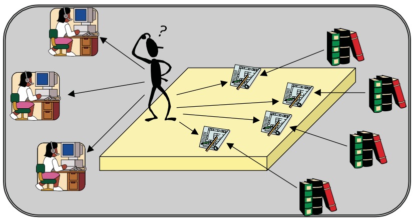


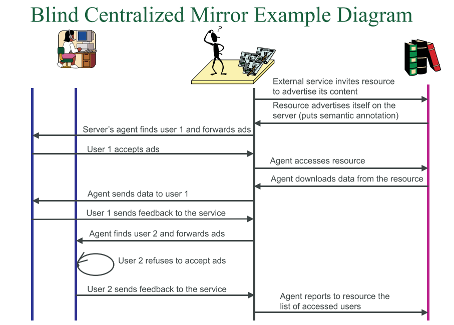
Architecture # 16: "Semantic Search from Server"



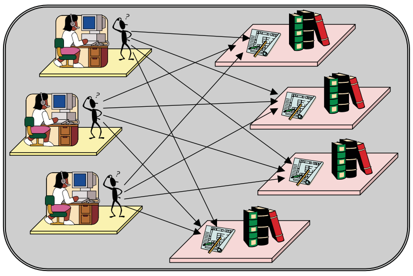


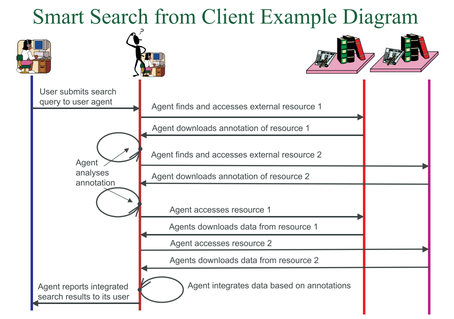
Architecture # 18: "Blind Centralized Mirror"



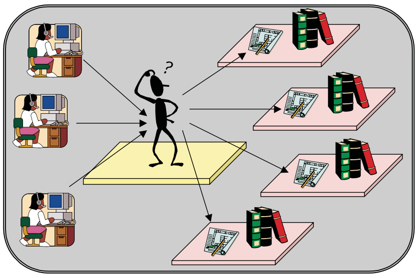


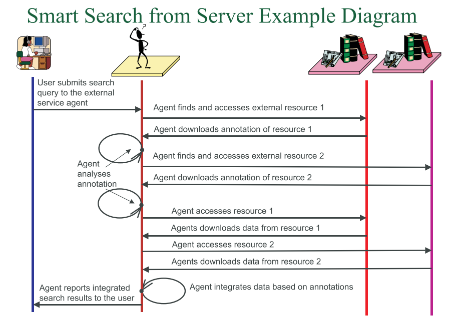
Architecture # 22: "Smart Search from Client"



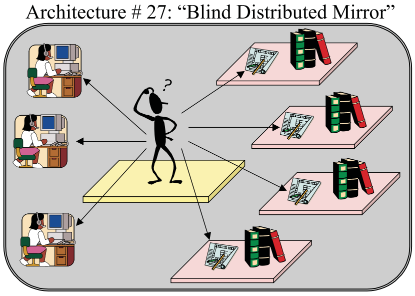


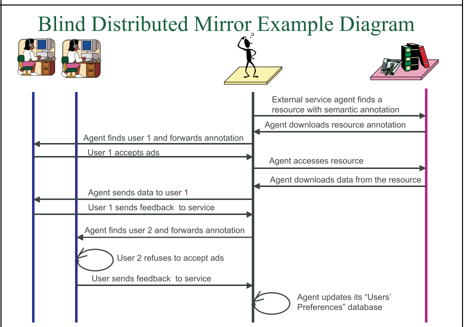
Architecture # 25: "Smart Search from Server"



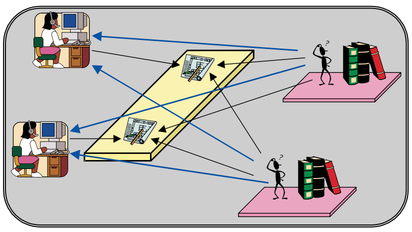


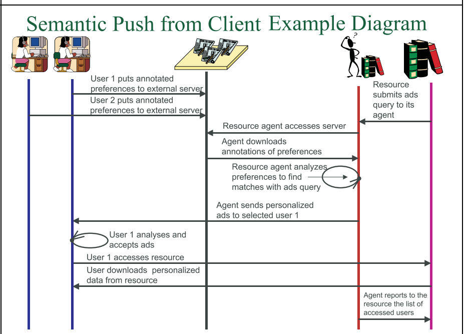
Architecture # 27: "Blind Distributed Mirror"



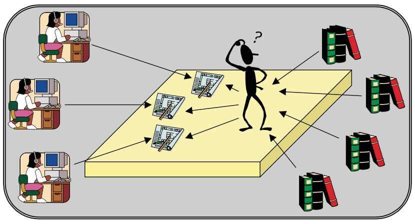


Architecture # 32: "Semantic Push from Client"

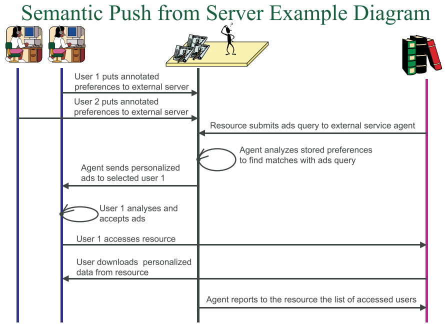




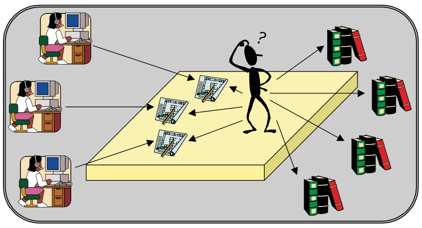
Architecture # 35: "Semantic Push from Server"



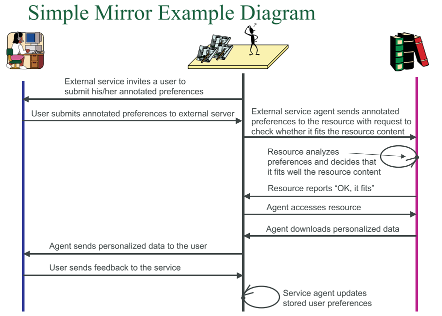
Semantic Push from Server Example Diagram



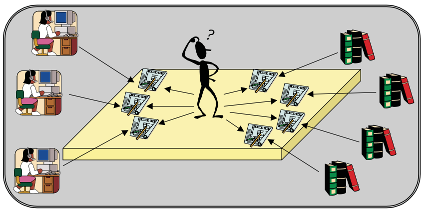
Architecture # 36: "Simple Mirror"



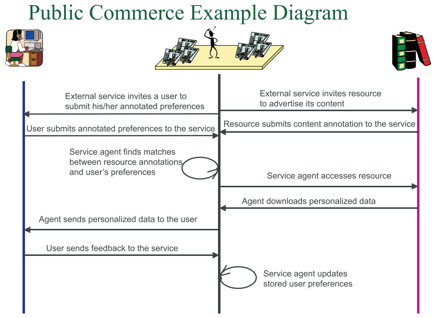
Simple Mirror Example Diagram



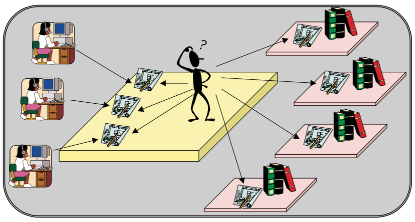
Architecture # 45: "Public Commerce"



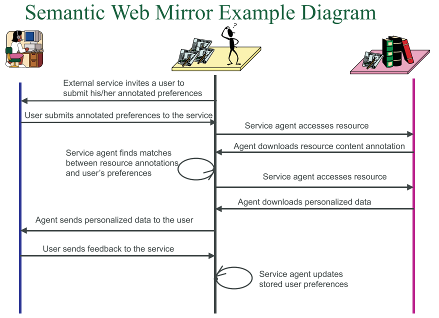
Public Commerce Example Diagram



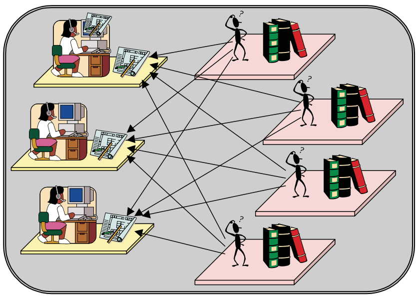
Architecture # 54: "Semantic Web Mirror"



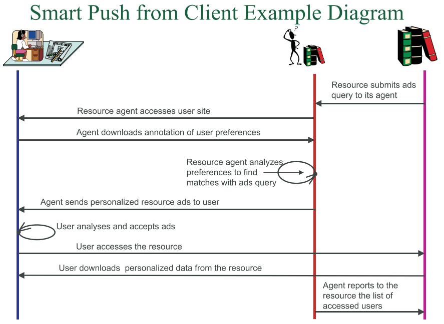
Semantic Web Mirror Example Diagram



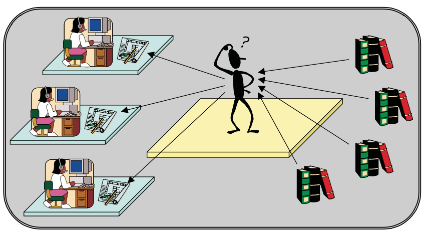
Architecture # 59: "Smart Push from Client"



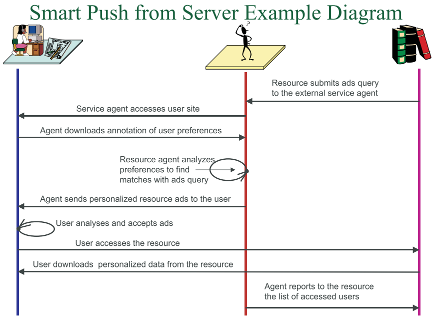
Smart Push from Client Example Diagram

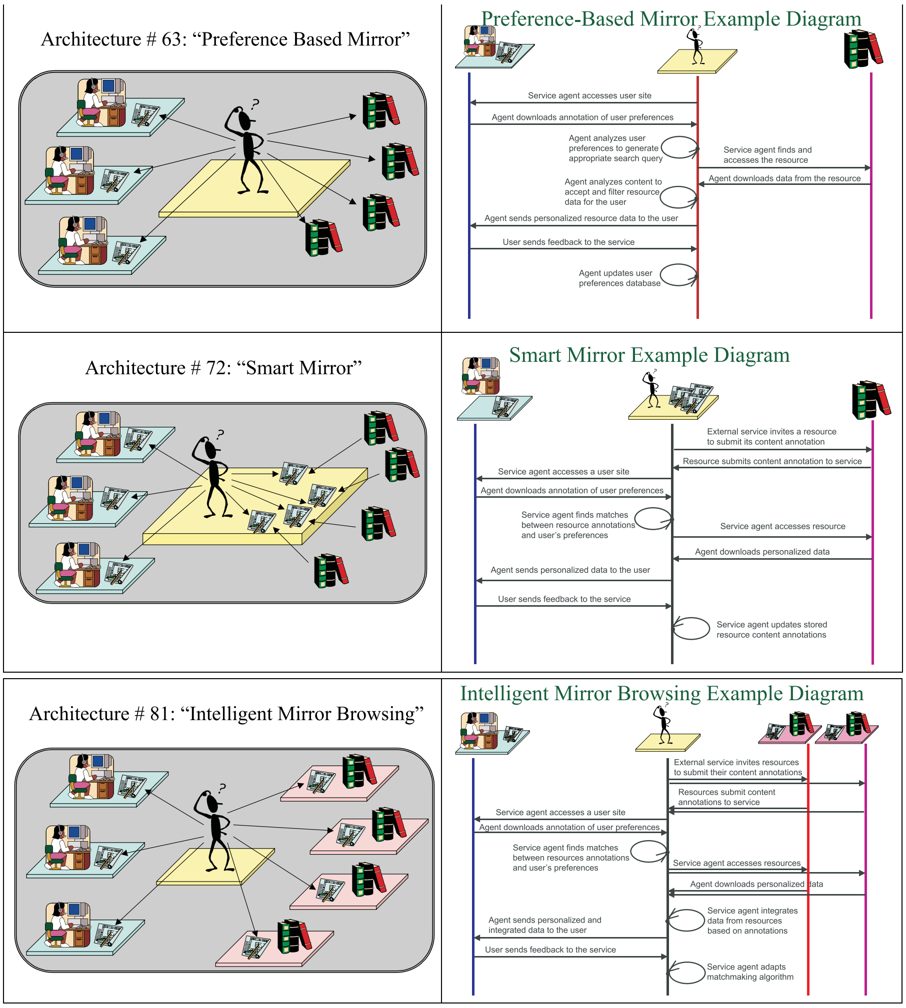


Architecture # 62: "Smart Push from Server"



Smart Push from Server Example Diagram





Vagan Terziyan (http://www.cs.jyu.fi/ai/vagan) received his Engineer's degree on Applied Mathematics in 1981 from Kharkov National University of Radioelectronics (KNURE). He became Candidate of Technical Sciences in 1985 on Natural Language Processing and Doctor of Technical Sciences in 1993 on Information Systems and Artificial intelligence. He is acting as Professor since 1994 and as the Head of the Artificial Intelligence Department since 1997 in KNURE. Area of research interests and teaching includes: Distributed AI, Semantic Web, Knowledge Discovery and Machine Learning, Mobile Electronic Commerce. He is Scientific Head of the Metaintelligence Laboratory in KNURE, Scientific Head of the Data Mining Research Group and Telemedicine Research Group, the Head of co-operation and exchange Program with University of Jyvaskyla (Finland) and similar Program with Vrije Universiteit Amsterdam (the Netherlands). He is KNURE's team leader in the OntoWeb IST Network Project. Recently he is working (in the areas of Semantic Web and Agent Technologies) as Associate Professor in MIT Department, University of Jyvaskyla and as Senior Researcher at the InBCT (Innovations in Business, Communication and Technology) TEKES Project in Agora Centre. He is also a visiting lecturer in Vrije Universiteit Amsterdam and Jyvaskyla Polytechnic. He has more than 100 scientific publications, about half of them in internationally recognised magazines and conferences.

References

1. V. Terziyan, Architecture for Mobile P-Commerce: Multilevel Profiling Framework, Proceedings of the IJCAI-2001 International Workshop "E-Business and the Intelligent Web", Seattle, USA, 5 August 2001, 12 pp.
2. D. Fentsel, Premises and Challenges of Research and Development in Information Agent Technology, available in: http://www.cs.vu.nl/~dieter/wgal/roadmap. html.
3. A. Whinston, D. Stahl, S. Choi, The economics of electronic commerce, Indianapolis (IN): Macmillan, 1997.
4. B. Lublinsky, Building a World-class e-Market, eAI Journal, August 2001, available in: http://www.eaijournal.com/PDF/LublinskyEMarket.pdf
5. G. Fischer, L. Palen, Knowledge Management - Pull and Push Technologies, Course description, University of Colorado, Spring 1999, available in: http://www.colorado.edu/~13d/courses/csci7212-99/pdf/knowledge-management.pdf.
6. T. Berners-Lee, J. Hendler, O. Lassila, The Semantic Web. Scientific American, 284(5), 2001, pp. 34-43.
7. D. Haverkamp, S. Gauch, Intelligent Information Agents: Review and Challenges for Distributed Information Sources, Journal of the American Society for Information Science, Vol. 49, No. 4, pp. 304-311, April 1998.
8. A. Maedche, S. Staab, Applying Semantic Web Technologies for Tourisntelligent Systems Magazine, January/February 2002, pp. 18-25.
10. B. Wondergem, P. Bommel, T. Weide, Cumulative Duality in Designing Information Brokers, In: G. Quirchmayr, E. Schweighofer, and T.J.M. Bench-Capon (eds.), Proceedings of the 9th International Conference on Database and Expert Systems Applications (DEXA'98), August 24-28, Vienna, Austria, pp. 125-134.
11. B. Hermans, Desperately Seeking: Helping Hands and Human Touch, 8 July 2000, Available in: http://www.hermans.org/agents2/index.html.
12. B. Wondergem, P. Bommel, T. Weide, Opportunities for Electronic Commerce in Agent-Based Information Discovery, In: F. Griffel, T. Tu and W. Lamersdorf (eds.), Proceedings of the Electronic Commerce Conference 1998 (TrEC98), Hamburg, Germany, June 1998, pp. 126-136.