Document brokering with agents: Persona approach

Junichi Suzuki and Yoshikazu Yamamoto *

Summary. The explosive growth of the Web places a new demand: how Web documents can be personalized for an each user. This paper addresses this demand and proposes an agent-mediated solution. Our system named Persona can provide appropriate presentations for different users or user groups on the fly. We also experimented an application to support the distributed software development using Persona. Agents make it easy to dynamically process documents and distribute them.

1 Introduction

It is now recognized that the Internet is a ubiquitous computing environment and huge distributed document repository. Various approaches to retrieve, collect or process documents on the Web have been proposed in both academic and commercial areas. In such a situation, there are two demands about the interface (or presentation) of Web documents:

- Device independent presentation and/or content
- Personalized document presentation and/or content

Currently, we can access Web documents with various devices: from personal electronic devices such as PDAs (Personal Digital Assistants), pagers and cellular phones into computers including parmtop, laptop and desktop ones. Most documents, unfortunately, are designed for the display on desktop computers that have color monitors with at least 640×480 resolution. The disconnect can lead to 4-to-1 (or greater) ratio of designed v.s. available screen area [1]. It is required to provide the mean to display arbitrary Web documents independently of the I/O capability.

The second demand involves the issue that every Web document has only one presentation and/or content across its entire readership. Obviously, it is a better solution to transfer documents adapted to each reader's own preference. Some search engines and publishers have started to provide such a capability [2, 3, 4]. The goal is to present documents for users in a timely appropriate and accessible manner. To achieve so-called document on demand capability described above, we have proposed a solution to the former problem, device independent presentation

* Department of Computer Science, Graduate School of Science and Technology, Keio University Yokohama City, 223-8522, Japan +81-45-563-3925 (Tel & FAX) suzuki@yy.cs.keio.ac.jp

and/or content [5]. Our system named Persona 1.0 runs on Web servers as a CGI (Common Gateway Interface) program and accepts the client-side environment information such as the resolution, extent of the current window and color/grayscale. It can re-author arbitrary HTML files in order to make their presentation suited to the client environment. The content re-authoring is done based on the rules defined by content providers.

In this article, we address the latter problem, personalized document presentation and/or content, and present our solution to personalize Web documents for individual users. Our revised Persona 2.0 system employs the agent technology to provide different presentations of XML (eXtensible Markup Language) documents for different users or user groups. Using the Persona 2.0 system, we have experimented an application to support the distributed software development as a proof-of-concept example.

The remainder of this paper is organized as follows: Section 2 introduces the concepts and characteristics of agent technologies. Section 3 outlines the conceptual framework of the Persona system. Section 4 describes design principles and implementation of Persona. Section 5 presents an application with Persona. In Section 6 and 7, we conclude with a note on the current status of the project and future work.

2 Agent technologies

In general, an software agent is a computational entity that:

- acts on behalf of its users or other entities in autonomous fashion (*agency* and *autonomy*).
- behaves with some level of *proactivity* and/or *reactivity*.
- exhibits some level of *learning*, *cooperation* and *mobility*.

Researchers and software companies have set hopes on agents, which know users' interests and can act to meet their goal autonomously without their supervision. Recent researches have identified three promising application domains: *intelligent interface agents*, *distributed agents* and *mobile agents* [6].

Intelligent (user) interface agent [7] intermediates users and their software system to provide the adaptive user interface for users. They can assist users by providing a simplified interface to a complex system, learning the user profile and changing presentation for a system's internal components.

Distributed agent (a.k.a. *multi agents*) means the agent that is not centralized to meet a goal. These agents originated in the research of *Distributed Artificial Intelligence*, which is a subfield of Artificial Intelligence (AI) and has generated several key technologies including neural network. The multi agent system can be defined as a loosely coupled network of problem solvers (i.e. agents) that work together to solve problems that are beyond their individual capabilities [8].

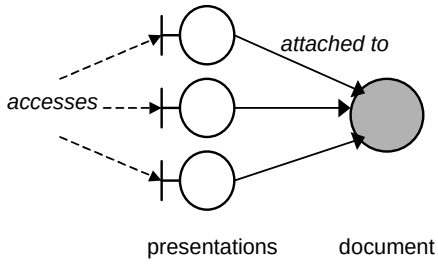

図 1. Multiple presentations can be attached to a single document. Users access the document through a selected presentation.

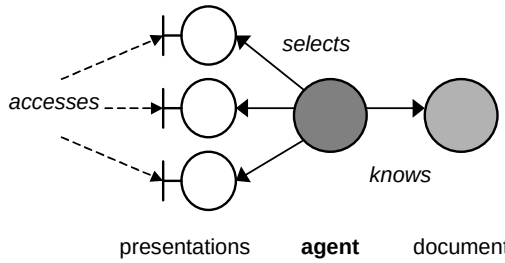

図 2. An agent intermediates between a document and its presentations. It adapts the presentation of a target document for a specific user.

Mobile agent [9] is a software entity that can migrate from machine to machine on the network to solve a given problem. It manages their own routing schedule, recovery and filtering behavior [10].

3 Conceptual framework of Persona

As described in Section 1, the goal of Persona is to provide Web documents with a right presentation to right users. This capability is important in the situation where the presentation (e.g. format) of documents should be varied for each user, each group of users or a context of users' behavior.

Documents can generally have multiple presentations, and users access them through a selected presentation (Figure 1). The presentation can be determined statically at the time when the document is authored, or dynamically after it is published. Also, the presentation may be selected automatically or manually. CSS (Cascading Style Sheet), XSL (eXtensible Style Language) and DSSSL (Document Style Semantics and Specification Language), that are stylesheet description languages for HTML, XML and SGML respectively, intend the static and manual selection. Persona is an execution engine to rearrange the presentation for Web documents such as HTML and XML dynamically and automatically. The selection

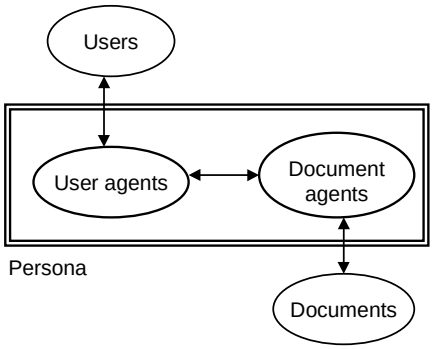

図 3. The interaction between users, agents and documents.

process is performed by agents called *document agents* (Figure 2). A document agent exists on per-document basis, and contains metadata of each document. In other words, a document agent knows the information on the target document and the possible selection of its presentations.

Every document agent needs to be personalized for a specific user or group of users in order to allows for the adaptation of documents. Therefore, a document agent have to know the information on users and groups of users. For higher levels of intelligence, it may have some representations of each user's interests, or some other understandings and reasoning about which presentation is best-suited to a user in a given context.

If document agents are created for each combination of a document and its user, a number of agents should exist. To reduce the number of agents and simplify the management of them, a *user agent* is introduced, which represents each user. Document agents and user agents communicate with each other as brokerages between documents and their users. Figure 3 shows the relationships between users, agents and documents.

Consequently, document agents and user agents in Persona are categorized as intelligent (user) interface agents described in Section 2. These agents should have the following characteristics:

- Agency and autonomy
- Reactivity
- Cooperation between agents

Issues in the agent-mediated adaptive presentation includes:

- how to prepare multiple presentations on the same document.
- how to prevent conflicts between multiple presentations.
- how to find and apply an appropriate presentation to a target document on the fly.

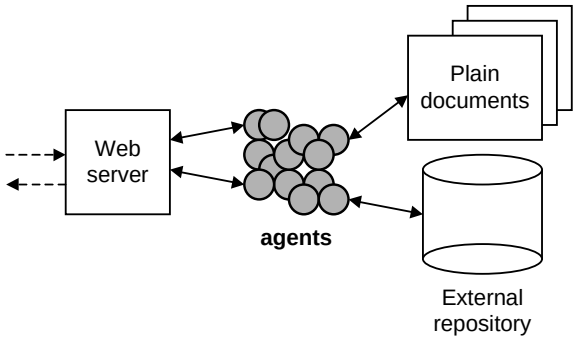

図 4. Agents receive a request through a Web server and control the documents.

- how to cooperate user agents with document agents.

4 Design and implementation of Persona

This section describes the design and implementation of Persona to achieve the concepts and principles introduced in Section 3.

4.1 Design

As depicted in Figure 4, Persona is intended to run at the back-end of Web servers. Once an user agent receives requests for a document and user information, it finds a document agent that represents the requested document and then asks it to choose an appropriate presentation. The current Persona system supports XML documents and XSL stylesheets. All presentations for XML documents are described with XSL. Generally, style engines, which apply a XSL stylesheet to a XML document, have some output formats such as HTML, RTF (Rich Text Format), LaTeX and PDF (Portable Document Format). In Persona, document agents output HTML formatted presentations because the system intends to use Web servers and HTML browsers. Document agents can retrieve a target document in a set of plain XML documents and external XML repositories (see Figure 4). They are created on per- document basis when a XML document is registered into a registry that holds the pairs of a directory path of the target document and reference to its document agent. User agents can find a document agent by referring the registry. Figure 5 shows the key interactions between components in Persona. The check-in manager initiates a document agent corresponding to the registered documents, and stores the document and its agent into a registry. The registry also contains a series of pairs of a user name and its agent.

Figure 6 depicts the class structure of core components in Persona. There is loose dependency between `UserAgent` and `DocumentAgent`. Each object communicates with each other through `Registry`. `Registry` contains zero or more instances

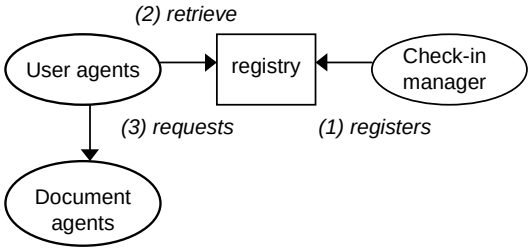

図 5. Key interactions among components in Persona.

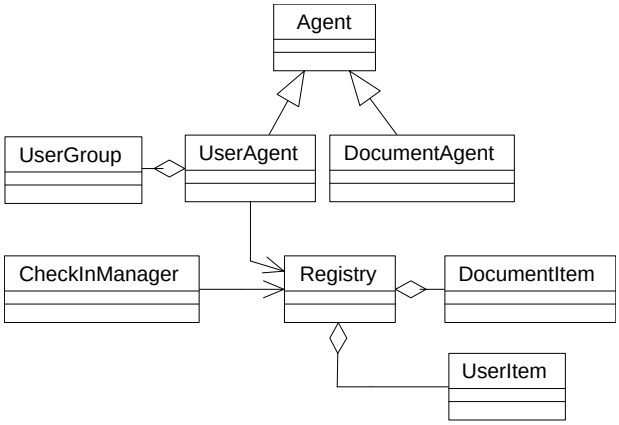

図 6. Class structure of core components in Persona.

of `DocumentItem`, which has pairs of the document path and its agent. It also has one-to-many relationship with `UserItem`, which contains pairs of a user name and user agents.

All agents are described with Persona Agent Description Format (PADF), which is based on XML. The check-in manager parses an agent definition written in PADF and instantiates either `UserAgent` or `DocumentAgent`. PADF can define the behavior of document agents; how to choose an appropriate presentation.

4.2 Implementation

Components in Persona can be implemented as a CGI program or server-side components like Servlet [11] (see also Figure 4 and 6). Our current system implements these components with Java and integrates them into OpenWebServer [12, 13], which is a reflective and extensible Web server.

Persona provides PADF DTD (Document Type Definition), which is a XML DTD. PADF DTD and sample descriptions can be obtained at [22]. Check-in manager, described in Figure 5 and 6, parses the PADF definition of agents. The definition have to be valid documents, which requires to contain (or link to) a

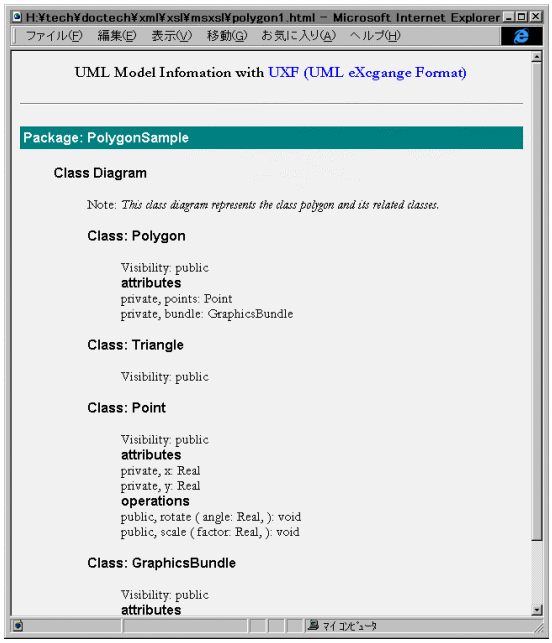

図 7. A sample textual presentation generated by Persona.

proper DTD and completely obeys its constraints. We are using a validating parser called nsgmls in creating PADF definition and non-validating parser called Lark in parsing it.

The most important task of `DocumentAgent` is to find and apply an appropriate presentation. It refers the PADF definition and and `UserAgent` to choose a best suited XSL definition. Then, it uses a XSL style engine called msxsl to apply a selected preesntation to the target XML document. A newly created HTML document is sent back to a HTML browser.

5 Application

This section presents an application with Persona. [14] lists seven areas where the agent technology is expected to contribute significantly; network management, e-mail, information management, teamwork, electronic commerce, mobile computer users and typical computer users.

We are interested in the areas of information management and teamwork. Agents could filter, condense and present the information along with rules specified by the users, also automatically forward documents, send reminder notices, schedule meetings, and keep track of the teamwork process. Using Persona, we extended our existing distributed system that shares and manages software design models within a project team [15, 16]. This platform was originally developed to interchange software models based on UML (Unified Modeling Language) [17, 18],

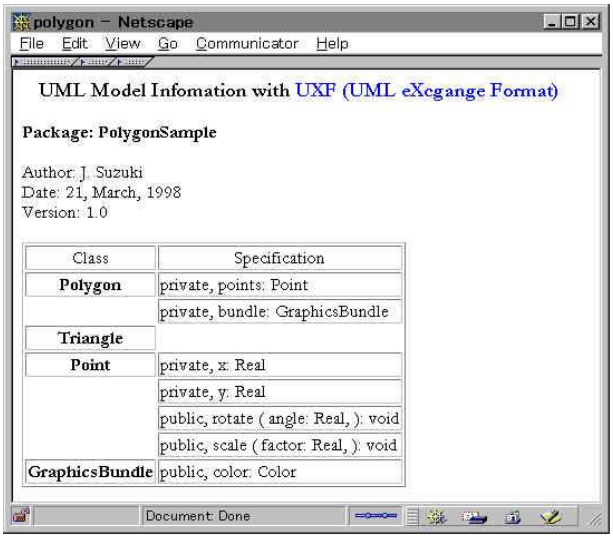

図 8. A sample table presentation generated by Persona.

which is a standard object-oriented analysis/design methodology. Our system uses UXF (UML eXchange Format) [19], which is a XML-based format to encode UML models in a text stream, and circulates design documents formatted in UXF within a development team. It leverages:

- Team development: Our system allows developers to continue their work concurrently at physically separated places. It eases a smooth progress of the development project.
- Reuse of development information: Our system allows the model information to be interchangeable between various development tools. This seamless interoperability increases our productivity of software modeling.

We extended this system that it can provide the adaptive presentation for UXF documents. As an example, we used a simple design model and its UXF representation . For this UXF (virtually XML) document, we prepared three presentations: a textual stylesheet, table stylesheet and Graphical Java applet. The textual presentation shows a textual sequence of the name of classes, their attributes and signatures of methods (Figure 7). The table presentation shows a table that is composed of classes and its specifications (Figure 8). The graphical presentation visually shows the relationships between packages and between classes (Figure 9). It can also present the specification of a selected class in right pane. This presentation is created using the ConceptMap applet [20]. Persona can selectively apply an appropriate presentation for individual users and user groups. PADF definition for this application is provided at [22].

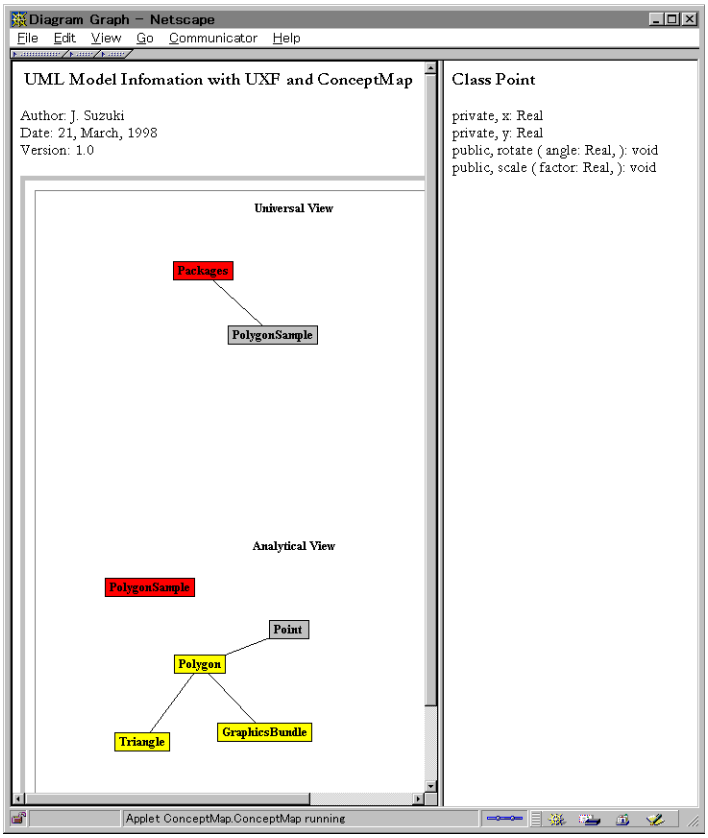

図 9. A graphical presentation using ConceptMap applet.

6 Discussion and Future work

As described earlier, the knowledge of agents is described with XML based format called PADF. World Wide Web Consortium (W3C) is now in the process of standardization of Resource Description Framework (RDF) [21]. RDF allows us to describe the metadata of Web documents and interchange its semantics across them. We are investigating RDF as a description language of document agents. We are developing an external repository like an object oriented database to store XML documents. It will be integrated with OpenWebServer (see also Figure 4). In addition, we are also working for a caching mechanism because we suffer from the latency due to the dynamic adaptation of presentations.

7 Conclusion

This paper addresses how the XML documents can be personalized for individual users, and proposes our agent- mediated solution. The Persona system can choose and provide an appropriate presentation of the target document for different users

or user groups. We also experimented an application to support the distributed software development using Persona. Agents make it easy to dynamically process documents and distribute them.

参考文献

[1] T. W. Bickmore. Digestor: Device-independent access to the World Wide Web. In *Proceedings of the Sixth International World Wide Web Conference*, 1997.

[2] Firefly. at http://www.firefly.com/.

[3] My Yahoo. at http://my.yahoo.com/.

[4] Interlligent Agent Group at Trinity College Dublin. Survey of Virtual Newspaper Systems. at http://www.cs.tcd.ie/research_groups/aig/iag/news.html.

[5] J. Suzuki, G. Tashiro, Y. Abe and Y. Yamamoto. Persona: adaptive presentation for Web documents. Submitted to *IPSJ Summer Programming Symposium '98*.

[6] S. Green et. al. Software Agents: A review. at http://www.cs.tcd.ie/research_groups/aig/iag/toplevel2.html.

[7] B. Laurel. Interface Agents: Metaphors with character. In Software Agents (M. Bradshaw, ed.), MIT Press, 1997.

[8] G. O'Hare and N. Jennings. (Eds). Foundations of Distributed Artificial Intelligence. John Wiley & Sons, 1996.

[9] J. E. White. Mobile Agents. In Software Agents (M. Bradshaw, ed.), MIT Press, 1997.

[10] L. Hurst et. al. Mobile Agents -Smart Messages. In *Proceedings of the First International Workshop on Mobile Agents*, 1997.

[11] JavaWebServer and Servlet. at http://jserv.java.sun.com/products/webserver/docs/.

[12] J. Suzuki and Y. Yamamoto. Building an adaptive Web server with a meta-architecture: AISF approach. In *Proceedings of JSSST SPA'98*, March, 1998.

[13] J. Suzuki and Y. Yamamoto. Experimental implementation of adaptive Web server using software patterns. Submitted to *the special issue on design patterns of IEEE Communications magazine*, 1999.

[14] L. L. Lensnick and R. E. Moore. Creating cool intelligent agents for the net. IDG Books, 1997.

[15] J. Suzuki and Y. Yamamoto. Managing the software design documents with XML. In *Proceedings of ACM SIGDOC '98*, 1998.

[16] J. Suzuki and Y. Yamamoto. Toward the interoperable software design models: quartet of UML, XML, DOM and CORBA. Submitted to *IEEE ISESS '99*, 1999.

[17] Rational Software et.al. UML Semantics. Object Management Group document number: ad/97-08-04, 1997.

[18] Rational Software et.al. UML Notation Guide. Object Management Group document number: ad/97-08-05, 1997.

[19] J. Suzuki and Y. Yamamoto. Making UML models exchangeable: UXF approach. In *Proceedings of UML'98*, 1998.

[20] J. Suzuki. Design and coding of ConceptMap. Submitted to Java Report, SIGS publication.

[21] O. Lassila and R. R. Swick (eds). Resource Description Framework (RDF) Model and Syntax Specification. W3C Working Draft 19 August 1998.

[22] Persona project Web site. http://www.yy.cs.keio.ac.jp/~suzuki/project/persona/.