Using Description Logics for the Provision of Context-Driven Content Adaptation Services — Source link ↗

Stephen J.H. Yang, Jia Zhang, Jeff J.S. Huang, Jeffrey J. P. Tsai

Institutions: National Central University, Northern Illinois University, University of Illinois at Chicago

Related papers:

- Multi-purpose Adaptation in the Web of Things

- The Adaptation of a Web Information System: A Perspective of Organizations

- Distributed Context Aware Dynamic Adaptation Model for Knowledge Assessment in E-Learning System

- Enrich the Expressiveness of Multimedia Document Adaptation Processes

- HaMA: A Handicap-based Architecture for Multimedia Document Adaptation

Using Description Logics for the Provision of Context-Driven Content Adaptation Services

Stephen J.H. Yang, National Central University, Taiwan

Jia Zhang, Northern Illinois University, USA

Jeff J.S. Huang, National Central University, Taiwan

Jeffrey J.P. Tsai, the University of Illinois at Chicago, USA

ABSTRACT

This paper presents our design and development of a description logics-based planner for providing context-driven content adaptation services. This approach dynamically transforms requested Web content into a proper format conforming to receiving contexts (e.g., access condition, network connection, and receiving device). Aiming to establish a semantic foundation for content adaptation, we apply description logics to formally define context profiles and requirements. We also propose a formal Object Structure Model as the basis of content adaptation management for higher reusability and adaptability. To automate content adaptation decision, our content adaptation planner is driven by a stepwise procedure equipped with algorithms and techniques to enable rule-based context-driven content adaptation over the mobile Internet. Experimental results prove the effectiveness and efficiency of our content adaptation planner on saving transmission bandwidth, when users are using handheld devices. By reducing the size of adapted content, we moderately decrease the computational overhead caused by content adaptation.

Keywords: Description logics; context-driven ; content adaptation; mobile Internet.

INTRODUCTION

Mobile computing poses big challenges to Web content delivery services in several significant ways. First, increasing volumes of handheld devices (e.g., Personal Digital Assistants (PDAs) and mobile phones) have been used to access Web content nowadays; however, most of the existing Web content is originally designed for desktop devices instead of handheld devices. Second, mobile users usually move constantly; their residing environments thus may change accordingly and Web content delivery should also subject to the changes for a better performance. For example, if a user moves into a blurred environment (e.g., due to a sunny or gloomy weather), the content should be consequently enlarged or the background color should be turned brighter. Third, people's status may change dynamically, which may consequently request adjusted content delivery. For example, if a user on a multimedia phone conversation walks into a room for another

physical meeting running in parallel, the audio should be turned off. The corresponding audio transmission thus becomes unnecessary.

Therefore, tools and mechanisms are in need to provide mobile users with transparent and seamless content delivery services. To achieve this ultimate goal, it is essential to deliver personalized and adaptive content according to users' situated environments. In this paper, the two terms "*situated environment*" and "*context*" are used interchangeably, both referring to content receivers' surrounding information that has an impact on content delivery and presentation such as receivers' personal profiles, receiving devices, communication network, location, activity, and time (Schilit, Adams et al. 1994; Dey & Abowd 1999; Satyanarayanan 2004; Mukherjee, Delfosse et al. 2005; Julien & Roman 2006).

The conventional approach to provide Web content supporting various types of computational devices is to prepare and maintain different versions (formats) of the same Web content for different devices. For example, a Web page typically holds one HTML version supporting desktop devices and one Wireless Markup Language (WML) version supporting wireless devices. This approach is straightforward but labor-intensive yet error-prone. Content providers have to prepare different layouts and formats for the same Web content, which results in tremendous overhead. To support a new device, all previous Web pages have to support a new format. Even worse, any change in the Web content may result in consequent changes in every related version, which is highly inflexible. Obviously, this approach is neither practical nor feasible for providers of a large volume of Web content.

To bridge the gap between content providers and mobile consumers, content adaptation refers to a technique that provides the most suitable content presentation by

means of transformation. While some researchers focus on content adaptation techniques on multimedia types, such as image and video adaptation (Mohan, Smith et al. 1999; Mukherjee, Delfosse et al. 2005; Nam, Ro et al. 2005; Vetro & Timmerer 2005; Xie, Liu et al. 2006; Wang, Kim et al. 2007), some other researchers focus on exploring how to conduct proper content adaptation based on receiving contexts (Lum & Lau 2002; Hua, Xie et al. 2006; He, Gao et al. 2007). Although the literature has witnessed these effective content adaptation efforts and techniques, they typically do not support automatic content adaptation decision; nor do they support configurable and extensible contextual environment specifications.

In contrast with the previous works lacking a clear semantic basis, this research intends to study a semantic foundation for content adaptation. Specifically, this research aims to investigate solutions to four research challenges:

> *C1: How to formally detect and represent mobile user contexts?*
> *C2: How to design configurable and reconfigurable adaptation rules?*
> *C3: How to manage and automate content adaptation management?*
> *C4: How to automatically generate adaptation output format?*

To the best of our knowledge, our research is the first effort to apply description logics (DLs) to formally define context profiles and requirements, and to automate content adaptation decision. We also introduce a Meta Medium Object concept and an Object Structure Model (OSM) to formally model an adaptable medium object. Then we construct a context-driven content adaptation planner that is designed and developed to automatically transform Web content to an appropriate format based on users' surrounding contexts,

especially when users are using handheld devices over the mobile Internet. Another benefit of our method it that it may reduce content access time. For example, if a user is accessing a film while driving, then our context-driven content adaptation planner automatically turns off the video for safety. This strategy could potentially save a significant amount of bandwidth by not transferring video clips (or other unnecessary data) in the already crowded mobile Internet.

The remainder of this paper is organized as follows. Related research regarding context and content adaptation planning is presented in Section 2. We formalize context profile and requirement definition using description logics in Section 3. We introduce a Meta Medium Object concept and an Object Structure Model (OSM) in Section 4. We present the design of or context-driven content adaptation planner in Section 5. Implementation considerations and solutions will be presented in Section 6. Experiments and evaluations are discussed in Section 7. We conclude the paper in Section 8.

RELATED WORK

He et al. (He, Gao et al. 2007) identify three types of objects in an HTML page, namely, structure, content, and pointer objects. Mukherjee et al. (Mukherjee, Delfosse et al. 2005) propose to associate content with metadata defining adaptation choices and their resulting media characteristics. In contrast with their work, we propose a concept of Meta Medium Object (MMO) as a self-containing entity encapsulating presentation objects and their metadata. Structural information of a Web page and the inter-relationships between presentation objects are managed by interactions and relationships between identified MMOs. Instead of having to manage two individual

objects for each content entity, our approach only has to manage one unified type of object. The maintainability and integrity become higher.

Hua et al. (Hua, Xie et al. 2006) propose a technique of supporting dynamic Web content adaptation based on semantic block identification, adaptation algorithms, as well as a caching strategy. While both partition a Web page based on the W3C's Document Object Model (DOM) (W3C) technique, our work differs from theirs after a DOM tree is constructed. Their strategy then tries to detect semantic blocks for page re-creation based on predefined receiving devices. In contrast, our approach then generates a lazy-initiated and dynamically managed Object Structure Model (OSM). In addition to supporting semantic block scaling (Hua, Xie et al. 2006), our research introduces the OSM as a flexible and adaptable foundational model to support context-driven content adaptation.

Adaptation rules, or policies, are typically used to guide content adaptations. Kinno et al. (Kinno, Yukitomo et al. 2004) adopt policy descriptions to designate how an adaptation engine should behave according to changing environments. Lemlouma and Layaida (Lemlouma & Layaida 2004) and Phan et al. (Phan, Zorpas et al. 2002), on the other hand, both design application-specific adaptation engines without employing adaptation policy descriptions. In contrast with their works based on predefined content adaptation rules, our research applies DLs to formalize context and requirement specifications and automate content adaptation decision making.

Mohan et al. (Mohan, Smith et al. 1999) introduce a rate-distortion framework equipped with a subjective fidelity measure for the analysis of adaptation policies. A representation scheme named InfoPyramid provides a multi-modal and multi-resolution representation hierarchy for multimedia. An

intelligent engine is introduced to help select the best content presentation style according to residing contexts, by modeling the selection process as a resource allocation problem in a generalized rate-distortion framework. Lum and Lau (Lum & Lau 2002) propose an adaptation system based on predefined decision policies. Upon receiving a request, the system first assigns a score to each possible content version based on various factors such as processing overhead and the volume of the content. Afterwards, a decision engine selects the optimal version by searching the score tree. In contrast with their work with high-level divisions of presentation qualities and immutable adaptation policies, our research proposes a fine-grained presentation Object Structure Model (OSM) supported by adaptable, configurable, and re-configurable adaptation policies.

Wang et al. (Wang, Kim et al. 2007) propose a utility function that considers video entity, adaptation, resource, utility, and their inter-relationships. Utility values are dynamically calculated to predict and prepare real-time video transcoding. In contrast with their work, we propose a rule-based decision engine equipped with configurable and re-configurable adaptation rules and lookup tables.

Many content adaptation prototypes have been built in recent years. Among them, Phan et al. (Phan, Zorpas et al. 2002) propose a middleware, called Content Adaptation Pipeline (CAP), to perform content adaptation on any complex data type, not only text and graphic image. XML is used to describe all the elements in a content hierarchy. Berhe et al. (Berhe, Brunie et al. 2004) present a service-based content adaptation framework. An adaptation operator is introduced as an abstraction of various transformation operations such as compression, decompression, scaling, and conversion. A logic adaptation path is determined by associating adaptation constraints to proper combinations of adaptation operators. To determine the optimal service, a path selection algorithm is proposed based on cost and time considerations. Their work shows a proof-of-concept of Web-based content adaptation; however, their implementations stay in a preliminary phase. How to map from constraints to adaptation operators is unsolved.

Lee et al. (Lee, Chandranmenon et al. 2003) develop a middleware-based content adaptation server providing transcoding utilities named GAMMAR. A table-driven architecture is adopted to manage transcoding services located across a cluster of network computers. Their approach allows incorporation of new third-party transcoding utilities. Lemlouma and Layaida (Lemlouma & Layaida 2004) propose an adaptation framework, which defines an adaptation strategy as a set of description models, communication protocols, and negotiation and adaptation methods. In contrast with their work, our content adaptation planner is founded on a theoretical model of OSM with an adaptation rule engine, as well as formal design patterns (Gamma, Helm et al. 1995).

Some researchers focus on content decomposition methods. Chen et al. (Chen, Xie et al. 2002a) propose a block-based content decomposition method, DRESS, for quantifying the content representation. An HTML page is factorized into blocks, each being assigned a score denoting its significance. DRESS selects the block with the highest score to represent the content. This method prevents from missing significant information. It also enables content layout to become adjustable according to the region of interest, attention value, and minimum perceptible size (Chen, Xie et al. 2002b). Ramaswamy et al. (Ramaswamy, Iyengar et al. 2005) propose an efficient fragment generation and caching method based on

detection of three features: shared behavior, lifetime, and personalization characteristic. The smallest adjustable element in these two approaches is a composite of objects (i.e., text, image, audio, and video). This granularity of decomposition is too large for mobile device screens; therefore, they are not suitable for mobile content adaptation.

Xie et al. (Xie, Liu et al. 2006) propose a decomposition and re-composition technique for facilitating browsing large pictures on small-screen devices. In contrast with their approach focusing on building a visual model of attention objects in an image, our research intends to establish a visual model of presentation objects in a Web page (i.e., HTML page).

MPEG-21 Digital Item Adaptation (DIA) adopts metadata to guide transformation of digital items for supporting various levels of Quality of Services (QoS) in accordance with a set of constraints such as storage, transmission and consumption (Vetro & Timmerer 2005). In contrast with their work focusing on generic content adaptation for providing different QoS features, our research focuses on Web page transformation oriented to mobile computing scenarios.

A number of context-based adaptation methods (Lei & Georganas 2001; Lum & Lau 2002; Pashtan, Kollipara et al. 2003; Toivonen, Kolari et al. 2003; Kurz, Popescu et al. 2004; Lemlouma & Layaida 2004; Krause, Smailagic et al. 2006) are proposed to customize Web content according to client contextual environments, including personal preferences, device capabilities, and access environments. Julien and Roman (Julien & Roman 2006) introduce a *view* concept to represent application-specific contextual information. Bellavista et al. (Bellavista, Corradi et al. 2003) adopt metadata for representing context characteristics at a high level of abstraction. Cabri et al. (Cabri, Leonardi et al. 2003) propose a

two-dimensional model to describe the location information in a mobile context: a physical location in space and a logical location within a distributed group or application. Nam et al. (Nam, Ro et al. 2005) propose to consider both user preferences and perceptual characteristics as a basis for effective visual content adaptation. Mukherjee et al. (Mukherjee, Delfosse et al. 2005) take into consideration run-time conditions about terminal, network, user preference, and rights. Krause et al. (Krause, Smailagic et al. 2006) believe that contextual information includes users states and surroundings. A user's state can be extracted from the user's activity, location, schedule, and physiological information. In contrast with their context models, in this research we propose a multi-dimensional context model. Along each dimension, users can configure attributes, each being associated with a list of configurable adaptation rules.

Some researchers focus on dynamic content adaptation based on users' changing contextual environments. CB-SeC framework (Mostefaoui, Tafat-Bouzid et al. 2003) and Gaia middleware (Roman, Hess et al. 2002) provide functionalities to help users find content based on their contexts or enable content adaptations according to user's contextual information. Mostefaoui and Hirsbrunner propose a formal definition of context to model content's contextual information (Mostefaoui & Hirsbrunner 2004) Lemlouma and Layaida propose a framework to assert metadata information of Web content (Lemlouma & Layaida 2001). All of these works use Composite Capabilities/Preferences Profiles (CC/PP) as interoperable context representation carriers to enable communication and negotiation of device capabilities and user preferences in terms of a service invocation. Zhang et al. further propose extensions to CC/PP to enable transformation descriptions between various

receiving devices (Zhang, Zhang et al. 2005). Besides, several Web Ontology Language (OWL)-based context models are presented (Khedr & Karmouch 2004; Yang, Zhang, Chen 2008; Yang, Zhang et al 2009) to provide high-quality results of Web content presentation beyond the expressive limitations of CC/PP. These researchers all utilize ontology to describe contextual information including location, time, device, preference, and network. In addition, some researchers pursue quantitative approaches to decide adaptation at run time. Among them, Wang et al. (Wang, Kim et al. 2007) formalize the decision process as a classification and regress problem. One major difficulty of these adaptation approaches is that content providers have to predefine exhaustive adaptation rules for adapting the contents for every contextual situation.

Another limitation of aforementioned content adaptation methods is that, they do not take users' changing contextual information into full account, especially when users are using handheld devices in a mobile environment. Users may receive some invalid content, and they have to manually verify received content. In contrast, our context-driven content adaptation rules can provide automatic and better contextually matched Web content to meet user' contextual changes in the mobile Internet.

As a fundamental technique in the field of semantic Web, Description logics (DLs) support formal knowledge representation in a structured manner (Baader, Calvanese et al. 2003). W3C-endorsed OWL (W3C 2004) is built on top of DLs. However, OWL is established to serve for generic Web services delivery. Compared to OWL, we apply DLs to study context-aware content adaptation. OWL can be used as one tool to specify contextual environments and content delivery requirements.

FORMALIZATION OF CONTEXT SPECIFICATIONS AND DECISIONS

To automate content adaptation decision, we apply description logics (DLs) to formally define context profiles and requirements.

Description Logics

Description logics (DLs) refer to a family of knowledge representation languages that can be used to express knowledge of an application domain in a structured manner (Baader, Calvanese et al. 2003). Equipped with formal logic-based semantics, DLs are capable of describing hierarchical notions of concepts (classes) and roles (relations) and formal reasoning about concepts and roles. It is a fundamental technique in the field of semantic Web; several widely used semantic Web languages are based on DLs such as W3C-endorsed OWL (W3C 2004).

Syntax of DLs contains three major components: predicate, relation, and constructor. A unary predicate symbol denotes a concept name; a binary relation denotes a role name; and a constructor is a recursive definition that defines comprehensive concepts and roles from atomic ones. In DLs, TBox (terminological box) is used to represent sentences describing concept hierarchies (e.g., roles between concepts); ABox (assertional box) is used to contain "ground" sentences stating to which in the hierarchy individuals belong (e.g., roles between individuals and concepts).

We adopt DLs to establish a hierarchical ontology to enable and facilitate context-aware dynamic content adaptation. Three layers are identified: a context requestor layer for describing receiving contextual environments, a context provider layer for defining contextual requirements and constraints, and a content planner layer for managing content adaptation definitions and

matchmaking.

Content Planner Layer

In the content planner layer, we define ContentPlanner as a superclass for content adaptation measurements and matchmaking. It has five subclasses formulated as follows: ContextProfile, ContextRequirement, ContextInquiry, AdaptationTemplate, and Metric.

$$ContentPlanner \subseteq T$$
$$ContextProfile \subseteq ContentPlanner$$
$$ContextRequirement \subseteq ContentPlanner$$
$$ContextInquiry \subseteq ContentPlanner$$
$$AdaptationTemplate \subseteq ContentPlanner$$
$$Metric \subseteq ContentPlanner$$

All content planner layer ontologies are described in DLs' TBox to formalize the descriptions and facilitate semantic matchmaking. ContextProfile defines the receiving contextual environments of the service requestor; ContextRequirement defines the required contexts of a content delivery service; ContextInquiry defines a service requestor's inquiry Ontology; and AdaptationTemplate defines some predefined content adaptation templates (e.g., in a format of tables or rules).

Metric defines an abstract template for the context requirement layer to properly define context attributes and their semantic meanings. For each attribute, an instance of the Metric class defines a 3-tuple (metricName, unit, value). The item metricName is a user-defined identifier for a metric. The item unit defines how to measure an attribute and implies its semantic meaning. For simplicity reason, we allow two types of unit: &xsd;#nonNegativeInteger and &xsd;#string (&xsd; is the entity macro

delimiting an XML Schema namespace). The former declares a numeric measurement as a cardinality constraint for a context attribute; the latter allows users to define application-specific measurements.

Metric is further divided into AtomicMetric and ComplexMetric: the former defines a metric over a single context attribute; the latter defines a metric over multiple context attributes through operators. The operands of an operator in a ComplexMetric definition can be either a single attribute or a composite attribute. An operator defines a function of how to process operand metrics. As a proof of concept, we define three operators: BooleanFunction, ArithmeticFunction, and AggregateFunction. A BooleanFunction allows three operations ($\vee, \wedge, \neg$). An ArithmeticFunction allows two operations (+,-). An AggregateFunction allows accumulating multiple metrics.

$$AtomicMetric \subseteq Metric$$
$$ComplexMetric \subseteq Metric$$

Each context metric is a subclass of either AtomicMetric or ComplexMetric. The taxonomy of metrics is designed by content service provider in the format of TBox; assertions on metrics are defined in the format of ABox. Proper definition of metrics is a key to appropriate content adaptation and delivery.

Context Requestor Layer

Without losing generality, in this research, we consider four aspects of a receiving context environment: receiver identity, receiving condition, network bandwidth, and receiving device.

Definition 1. The context profile of a receiver (ContxtProfile) is denoted as a 4-tuple:

ContextProfile = <I, C, N, D>, where:

I denotes the receiver's identity; C denotes the receiver's access condition; N denotes the receiver's communication network; and D denotes the receiver's receiving device. In the context requestor layer, we define ContextProfile as a superclass defining receiving contextual environments. ContextProfile has four subclasses: Identity, Condition, Network, and Device.

$$\text{ContextProfile} \subseteq T$$
$$\text{Identify} \subseteq \text{ContextProfile}$$
$$\text{Condition} \subseteq \text{ContextProfile}$$
$$\text{Network} \subseteq \text{ContextProfile}$$
$$\text{Device} \subseteq \text{ContextProfile}$$

Each dimension of ContextProfile is further refined into contextual attributes at a finer granularity. For example, as shown below, condition information can be further described by when, where, and what activities a person is involved, as well as the location identified by Global Positioning System (GPS), sensor networks, Radio Frequency Identification (RFID), and so on. Network information can be further described by communication protocols such as General Packet Radio Service (GPRS), Third-Generation Technology (3G), Voice over Internet Protocol (VoIP), Wireless Fidelity (WiFi), and so on.

$$\text{What} \subseteq \text{Condition}; \text{Where} \subseteq \text{Condition}; \text{What} \subseteq \text{Condition}; \text{Location} \subseteq \text{Condition};$$
$$\text{CommunicationProtocol} \subseteq \text{Network};$$

Furthermore, device information can be further described using Composite Capability/Preference Profiles (CC/PP), User Agent Profile (UAProf), and so on. Please note that the assumed information discussed in this paper is selected for ease of illustration; the actual contextual information in the real world could be much more complicated.

Context Requirement Layer

This layer defines suitable contexts for a content delivery service by the corresponding service provider. Existing DL reasoners typically provide better support for subsumption reasoning in TBox than datatype reasoning in ABox (Zhou, Chia et al. 2005); therefore, we adopt attribute-oriented cardinality to define contextual requirements and constraints. Note that contextual attributes are domain specific; it is up to content delivery providers to establish the attributes and cardinality specifications.

From content adaptation perspective, context requirements can be viewed as context input (ContextInput), and receiving context definitions can be viewed as required context output (ContextOutput). In addition, some other constrains may be defined. Some content might require external conditions to be satisfied to ensure that it can be properly displayed. For example, a video clip may require that a specific browser or software is installed for proper display. Such a specification is defined in ContextPrecondition. Furthermore, some side effects might be stated in (ContextEffect). For example, the display of some video clip might have the effect of lower throughout. Moreover, we use ContextDefault to represent no contextual requirements defined.

Thus, we define five ContextProfile classes for context requirements: ContextDefault, ContextInput, ContextOutput, ContextPrecondition, and ContextEffect. They are expressed as follows:

ContextDeFault $\subseteq$ ContextProfile

ContextInput $\subseteq$ ContextProfile

ContextOutput $\subseteq$ ContextProfile

ContextPrecondition $\subseteq$ ContextProfile

ContextEffect $\subseteq$ ContextProfile

In this layer, content service providers define the metrics for required context attributes. Supported by the definitions of the context attributes, cardinality constraints are ready to be added on the defined attributes in context planner layer for matchmaking purpose.

Our context ontology allows extensibility. Context attributes can be customized and specified. New attributes can be defined; existing attributes can be updated; and constraints on attributes can be specified and updated in context metrics hierarchy taxonomy. The metrics definitions and verifications in ABox are left to the corresponding domain and organization. Proper definition of this context requirement layer is the key to context matchmaking and content adaptation.

An Example Of Using DIs To Define Context Requirements And Environments

In this section, let us take access condition as an example to show how we apply DLs to define context requirements and environments.

As shown in Figure 1, a metric is defined to describe an attribute "condition" in ContextProfile. According to the abstract template described in section 3.2, condition is defined as a ComplexMetric:

ComplexMetric(condition)   :=
("condition", &xsd;#string, value)

Figure 1. Part of context ontology.
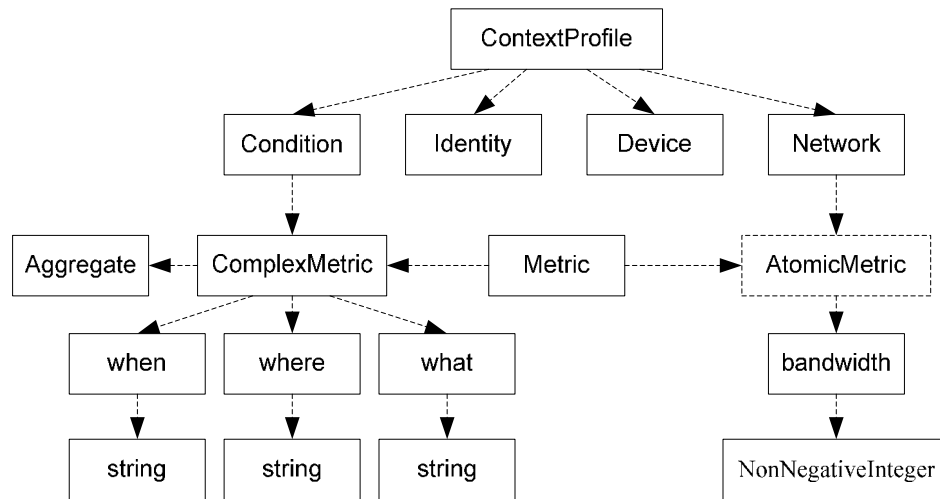
Condition is defined to describe people's access condition including when, where, and what activities they are involved. In other words, condition is a composite attribute that comprises three atomic attributes, each being defined using a string ("when", "where", and "what"). The three atomic attributes are aggregated into a composite attribute. For example, if a person intends to access films when she is involved in an activity "*meeting*"

(what) on "03/09/2008" (when) at a place "*meeting room A306*" (where), the receiving condition is defined as ("Condition", (when, where, what), (meeting, 03/09/2008, meeting room A306)). Using DL syntax, the profile can be represented as follows:

aProfile = ContextProfile

$\bigcap$ ("*meeting*" *what.WhatStringMetric*)
$\bigcap$ ("03/09/2008" *when.WhenStringMetric*)
$\bigcap$ ("*meeting room A306*" *where.WhereStringMetric*)

As shown in Figure 1, an AtomicMetric is defined for the context attribute *network* as follows:

AtomicMetric(network) := ("bandwidth", &xsd;#NonNegativeInteger, value)

A network connection can be measured by its bandwidth. For example, using DL syntax, a content service provider may define a suitable network connection as a bandwidth 400KB and 2.5MB in the following TBox definition:

aService = ContextRequirement

$\bigcap$ (≥ 400 *bandwidth.BandwidthKBMetric*)
$\bigcap$ (≤ 2500 *bandwidth.BandwidthKBMetric*)

The following ABox definition declares an assertion of such a context requirement published on 03/08/3008. The first line is a concept assertion; the rest lines are role assertions.

aService(aServiceInstance)
hasServiceProfile(aServiceInstance, "&provider_service;#")
startTime(aServiceInstance, 03/09/2008)

endTime(aServiceInstance, 03/09/2008)

Content Adaptation Decision

By applying DL-based ontology, the published requirements of a content delivery service define a set of constraint-specified context metrics in the format of ABox; the profile of a receiving party defines a set of constraint-specified context metrics in the format of TBox. Whether content adaptation is needed can be decided by comparing the two sets of specifications, which can be formalized into a problem of deciding ontology subsumption relationship. To automate the decision process, we define the following operator.

Definition 2. An isStronger operator ($\succ$) is denoted as an order operation between two DL specifications. For two constraints, x and y, on the same context attribute i, $x(i) \succ y(i)$ indicates that x(i) is a stronger specification than y(i).

Recall that we define two types of units for metrics: &xsd;#nonNegativeInteger and &xsd;#string. For the first type of unit, the applied domain is nonnegative integer that is an ordered set, meaning that any two values using this type of measurement unit are comparable. A general definition of a constraint (x) on a context attribute (i) specifies a range of variable values: $a \leq x(i) \leq b$, where $a \leq b$, and a and b are both non negative integers.

Axiom 1. For two constraints, x and y, on the same context attribute i, $a \leq x(i) \leq b, c \leq y(i) \leq d, a \leq b, c \leq d$, $x(i) \succ y(i)$, iff $c \leq a$ and $b \leq d$.

For simplicity, we consider that one constraint on a context attribute only specifies

one range. For the second type of unit (&xsd;#string), the applied domain is string. A general definition of a constraint (x) on a context attribute specifies an enumeration of string values: $x(i) \in A$, where $A = \{a_1, a_2, ... a_m\}$, and $a_l (1 \le l \le m)$ is a string. We define an isStronger operator ($\succ$) between them as follows:

Axiom 2. For two constraints, x and y, on the same context attribute i, $x(i) \in A, y(i) \in B, A = \{a_1, a_2, .. a_m\}, B = \{b_1, b_2, .. b_n\}$, $x(i) \succ y(i)$, iff $\forall a_k \in A, \exists p \Rightarrow a_k = b_p$.

Based on our definition of $\succ$ operator between two constraints on the same context attribute, we can define a compatible relationship ($\triangleleft$) between two context ontology descriptions, R and P (R is a content delivery service requirement, P is a receiving profile) as follows:

Definition 3. For $P = x(C) \bigcap x(D) \bigcap x(N), R = y(C) \bigcap y(D) \bigcap y(N)$, $P \triangleleft R$, iff $(x(C) \prec y(C)) \wedge (x(D) \prec y(D)) \wedge (x(N) \prec y(N))$

This means that unless a receiving profile is compatible with the required contexts, content adaptation is needed. Thus, we transform the problem of context constraint comparison into the problem of judging ontology subsumption relationship. The two operators both exhibit transitive properties, which are useful in content adaptation decision making.

Proposition 1. The isStrong operator $\succ$ has transitive property:
- $(x(i) \succ y(i)) \wedge (y(i) \succ z(i)) \Rightarrow (x(i) \succ z(i))$

Proposition 2. The compatible operator $\triangleleft$ has transitive property:
- $(P \triangleleft R) \wedge (R \triangleleft Q) \Rightarrow (P \triangleleft Q)$

Proof: These propositions are straightforward derived from their definitions.

Content Adaptation Rules

If a context comparison concludes a required content adaptation, an adaptation strategy is needed comprising a set of adaptation rules that guide proper adaptation of contents to fit users' contexts. We formally define an adaptation strategy as follows:

Definition 4. $AS_{personID} = \prod_{i=1}^{n} AR_i$, where:

AR_i is an adaptation rule, $i \in [1, n]$, n is the number of dimensions defined in the user's situated environment profiles.

For example, if people access films when they are involved in an activity "*meeting*" at a place "*meeting room A306,*" then the content adaptation should automatically turn off the sound to avoid making noise. If people access films when they are involved in an activity "*driving*" at a place "*car,*" then the content adaptation should automatically turn off the video for their safety. Surely, people could disable content adaptation if they decide to do so. For example, they may insist on listening to content during meetings (e.g., via ear phones), or reading content while they are driving (e.g., by stopping the car and read).

We formalize an adaptation rule as an event controller that automatically responds to changes of users' situated environments.

Definition 5. An adaptation rule can be represented by a 6-tuple as follows:

r = <id, o, r, p, g, tS>, where:

id denotes the identifier of the rule; *o*

denotes the original content; *r* denotes the resulting content; *p* denotes the prerequisite of the rule; *g* denotes the objectives of the rule; *tS* denotes the transformation strategy of the rule.

We use the following two rules as examples to explain the major components of a content adaptation rule.

Rule#	original object	Network	adapted object
N1	Object(video,original)	Network(2M+)	Object(video,original)
N2	Object(video,original)	Network(2M-) and ObjectSize(2M+)	Object(video,low_resolution)

The two rules can be represented as:

r1 = <"N1", Object(video.original), Object(video.original),
"ContextRequirement $\bigcap (\geq 2bandwidth.BandwidthM\,BMetric)$, null, null>

r2 = <"N2", Object(video.original), Object(video.low_resolution),
"ContextRequirement
$\bigcap (< 2bandwidth\,.BandwidthM\,BMetric\,) \bigcap (\geq 2objectSize\,.ObjectSize\,MBMetric\,)$",
"Lower video resolution", "videoTransMethod">

The first rule (N1) indicates that if the network condition is fast enough (>2M), video content does not have to be transformed. The second rule (N2) indicates that if the network condition is slow (a TBox definition <2M) and a video object is large (a TBox definition >2M), then the video object should be transformed into a lower resolution using the specified method (videoTransMethod).

All rules can be stored in a rule base incrementally: new facts can be asserted when new contextual information is acquired; new adaptation rules can be inserted when new context types are considered. Meanwhile, facts and rules can be removed when they are no longer suitable in a user's situated environment. Any insertion and deletion of facts and rules to an existing rule base may result in inconsistency or conflicts. Therefore, truth maintenance is required to ensure the consistence of the rule base. We have developed a truth maintenance mechanism for verifying rule base concerning redundancy, inconsistency, incompleteness, and circularity to ensure the incrementally constructed rule base is sound and complete (Yang, Tsai et al. 2003).

META MEDIUM OBJECT

Presentation content typically comprises different types of presentation objects, in the format of text, video, audio, image, and so on. In this section, we will focus on transforming presentation objects in different medium types; therefore, throughout the rest of the paper we will use the terms "presentation objects" and "medium objects" interchangeably. Our previous work yields algorithms to automatically identify semantically coherent presentation units (Unit of Information or UOI) in a Web page that have to be displayed on the same screen (Yang, Zhang et al. 2007). Each UOI may also comprise different medium objects. Since different transformation strategies and procedures are needed to serve different medium types, our adaptation process is focused on various medium objects.

MMO Concept

To formalize the content adaptation process, we introduce a concept of *MetaMediumObject* (MMO) that represents a self-contained meta concept of a presentation object encapsulating various presentation formats controlled by adaptation rules. As a proof of concept, we consider two major aspects of presentation formats: modality and fidelity. Modality indicates the type of a medium object such as text, video, audio, and image. Each modality is associated with a certain fidelity indicating the quality of the presentation of the objects, such as image resolution, color depth, and video bit-rate. Thus, a content adaptation rule can be viewed as a pattern for transforming a

medium object's modality and fidelity. We also simplify the description of modality and fidelity in this paper. In the real world, modality information can be further classified into flash or streaming media, for example; fidelity information can be further described by image types (e.g., BMP or JPEG), image resolution (e.g., 24 bit), color depth (e.g., 32bit), video bit-rate, and so on.

Figure 2 illustrates our idea of MMO by using a UML Class Diagram. An MMO represents an individual presentation object. Instead of being a presentation object with actual code, it is a meta-object containing the high-level information of the described presentation object.

Figure 2. Concept of MetaMediumObject (MMO).
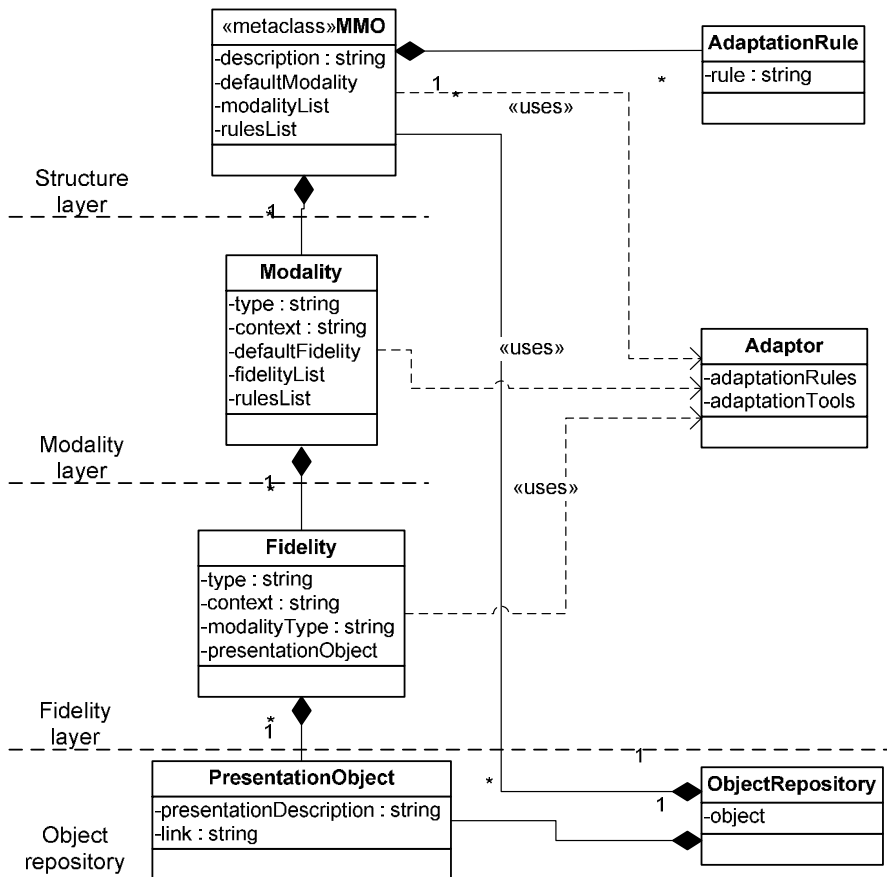
Definition 6. An MMO is denoted as a 4-tuple:

aMMO = <d, dM, M, R>, where:

d contains the basic descriptive information of the object such as its goal and short description; *dM* specifies the default modality of the presentation object. Without losing generality, the value of *dM* is one of the four items from the enumeration (text, image, audio, video): $dM \in \{t,i,a,v\}$. *L* contains a list of possible modalities in which the object can be represented. For example, a text object can also be represented as a piece of audio: $L = \{t,a\}$. As shown in Figure 2, an MMO may contain multiple modality objects. *R* contains a list of content adaptation rules associated with the presentation object for modality transformation, for example, whether the presentation object is allowed to be transformed from text to audio.

As shown in Figure 2, each modality object is a 5-tuple:

m = <t, c, dF, F, R>, $\forall m \in M$, where:

t indicates the presentation type of the presentation object, which is one of the four items from the enumeration (text, image, audio, video): $t \in \{t,i,a,v\}$. *c* indicates the contexts for which the modality is appropriate: $c \in TBox$. *dF* specifies the default fidelity of the presentation object for the associated modality. *F* contains a list of possible fidelities in which the object can be represented. For example, an image object can be represented in two resolutions, either JPEG 1024x728 or BMP 16-bits. *R* contains a list of adaptation rules associated with the presentation object for fidelity transformation, for example, whether an image object is allowed to be transformed from format JPEG 1024x728 to BMP 16-bits. Again, a modality object may contain multiple fidelity objects.

As shown in Figure 2, each fidelity object is a 4-tuple:

f = <t, mT, c, o>, $\forall f \in F$, where:

t indicates the user-specified fidelity type. *mT* indicates the category of presentation modality, which is one of the four items from the enumeration (text, image, audio, video): $mT \in \{t,i,a,v\}$. *c* indicates the contexts for which the fidelity is appropriate. *o* points to the actual presentation object.

As shown in Figure 2, an independent component *adaptor* is used by MMOs, modality objects, and fidelity objects to transform an existing presentation format into another. An *adaptor* typically is a tuple: (R, T). *R* define configurable and re-configurable adaptation rules for the adaptor; *T* links to a set of available content adaptation tools (e.g., text-to-audio tool and audio-to-text tool). Such a centralized adaptor allows us to build a system-level facility to manage configurable and reconfigurable content adaptation based on rules and patterns.

MMO Management

We establish a dedicated three-layer object structure model (OSM) to manage MMOs for presentation adaptation and version control. OSM can maintain and manage all possibilities of modality and fidelity associated with each presentation object.

Definition 7. An OSM is denoted as a 5-tuple:

OSM = <MMOs, M, F, R, O>, where:

MMOs indicate a set of MMOs; *M* indicate a set of modality objects associated with the MMOs; *F* indicate a set of fidelity objects associated with the MMOs; *R* indicate a set of content adaptation rules associated with the MMOs; *O* indicate a set of actual

presentation objects associated with the MMOs.

As shown in Figure 3, OSM implies a three-layer framework: structure layer, modality layer, and fidelity layer. The structure layer holds a tree-like structure of MMOs decomposed from a Web page. For each MMO (leaf node) in the structure layer, each of its modality attributes is mapped into a child node in the modality layer. For each modality node in the modality layer, each of its fidelity attributes is mapped into a child node in the fidelity layer. Each fidelity node maps to an actual presentation object (i.e., adapted version according to specified attributes).

Figure 3. Object structure model.
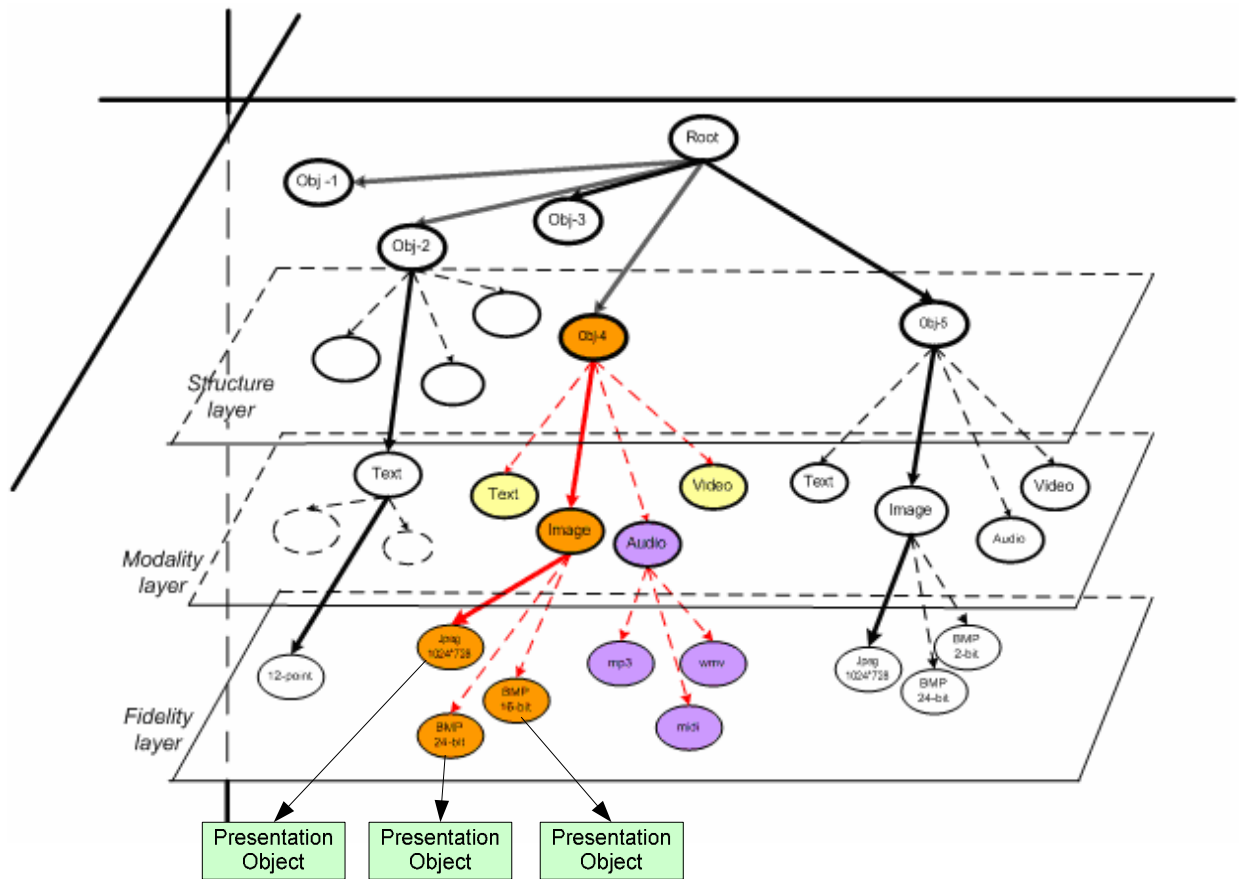
Each Web page can have an instance of OSM associated with it, representing in a tree-like structure. Figure 3 shows such an illustrative OSM tree. Taking *Obj4* as an example, it can be represented in one of the four types of modality (text, image, video, and audio); its modality of image supports three types of fidelity (JPEG 1024x728, BMP 24-bits, and BMP 16-bits). Therefore, *Obj4* in the structure layer is associated with four child nodes in the modality layer, in which *Image* node is further associated with three child nodes in the fidelity layer. Each fidelity node eventually points to a presentation object. *Obj4* and all of its descendent nodes form an object cluster that refers to a sub-tree of the

OSM, indicating all possible combinations of modality and fidelity associated with the object (*Obj4*).

CONTEXT-DRIVEN CONTENT ADAPTATION PLANNER

Based on our DL-based context formalization and MMO presentation, we have designed a rule-based planner for supporting and automating context-driven content adaptation, as presented in Figure 4. The design of the planner comprises five major components: (1) Web page decomposition, (2) object management, (3) dynamic content adaptation including dynamic transcoding and cache management, (4) adaptation planning, and (5) layout template composition. Each component also represents a step (phase) in the corresponding content adaptation process. As shown in Figure 4, seven database repositories are included in the planner: original Web pages, layout template repository, profile repository, adaptation rules, object repository, cache, and adapted pages. We will discuss the five components together with their associated repositories (highlighted in different colors) in detail in the following sections.

Figure 4. Design of content adaptation planner
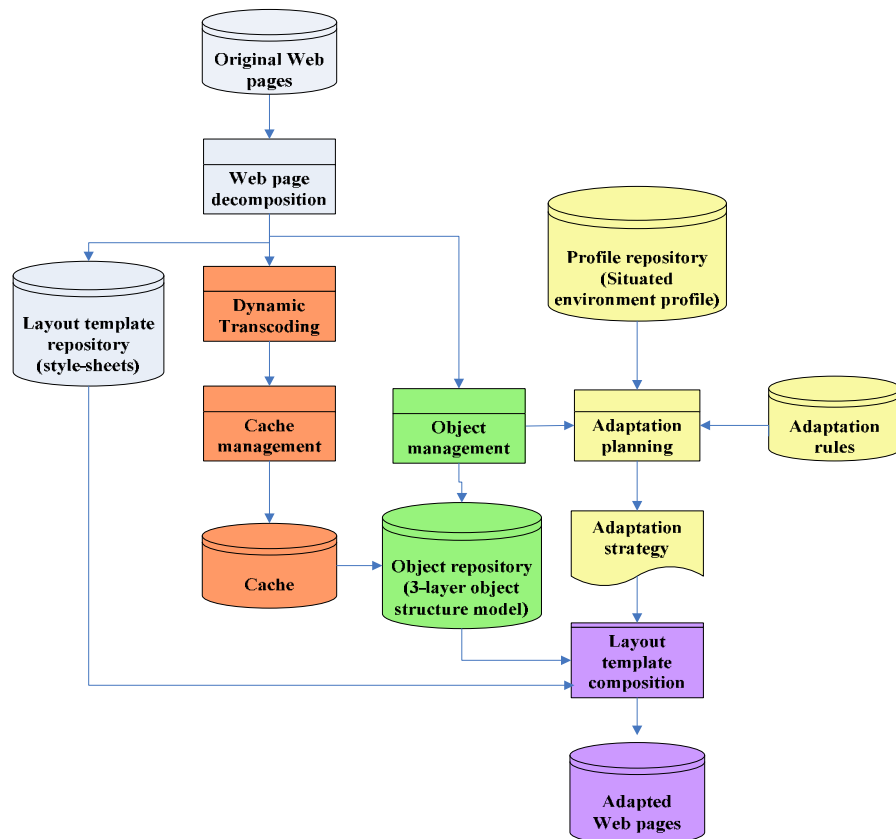
Web Page Decomposition

A Web page typically comprises a set of inter-related presentation objects, or simply

16

objects. Objects on a Web page are characterized by their modality indicating their types such as text, video, audio, and image. Each modality is associated with fidelity indicating the object's presentation quality such as image resolution, color depth, and video bit-rate. In order to render the same object on various devices, content adaptation may have to perform transcoding and change object's modality and fidelity accordingly. For example, if a mobile phone can only play images with a low resolution, the fidelity of an image with a high resolution has to be adapted to a lower level. In this paper, we focus on the design of content adaptation by providing rules for transforming objects' modality and fidelity to fit in with users' situated environment.

The input of the page decomposition step is a Web page in HTML format from the *original Web pages* repository. The goal of this step is to decompose the HTML page into encompassed presentation objects. To enable automatic page decomposition, a non-well-formed Web page has to be first transformed into a well-formed format. We adopt an open-source software tool Tidy (Bose & Frew 2005)to perform the task due to its simplicity and our familiarity with it.

Then we decompose the formatted Web page by identifying presentation objects based on W3C's Document Object Model (DOM) (W3C). As an *ad hoc* standard for Web components, DOM proposes a platform- and language-independent model representing the content, structure, and style of documents. The resulting objects are stored in the *object repository* through the *object management* phase.

Besides identifying individual objects as instructed by the DOM framework, we also detect and identify the inter-object relationships during the Web page decomposition phase. Relationships, such as spatial and temporal relationships, can be used to describe or decide the layout and presentation sequence of objects during Web page rendering and re-rendering. For example, for two objects that are rendered side-by-side spatially in a desktop's screen, they could be adapted to be displayed sequentially from top to bottom on a mobile phone's screen.

In order to maintain inter-object relationships, we construct an object tree for each decomposed Web page and record encompassed objects' spatial and temporal relationships in the tree. Figure 5 illustrates how a commonly accessed Web page (http://www.yahoo.com) is transformed into an object tree. As shown in Figure 5, objects can be displayed individually or aggregated into segments. The root of an object tree represents the Web page; the intermediate nodes represent segments; and the leaf nodes represent individual presentation objects. The details of how to generate such an object tree will be discussed in later sections. An object tree can be traversed in any order, either pre-order or post-order, where a node is accessed before any of its child nodes or after any of its child nodes. The resulted linear sequence will be used to restore inter-object relationships when we perform page layout template composition at a later stage. It can also be used to serialize the structure into the *object repository* to save memory space.

Figure 5. Example of a decomposed Web page and its corresponding object tree.
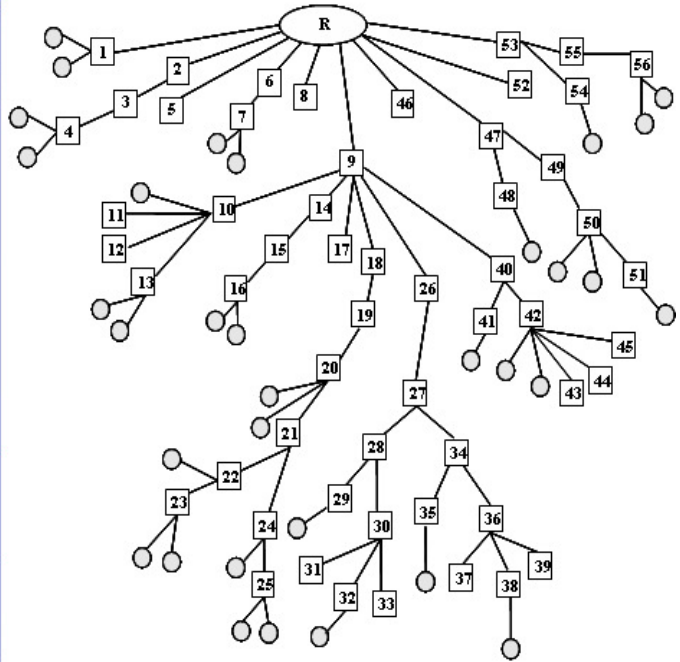
OSM And MMO Construction

After presentation objects are extracted from an HTML page, we can establish an OSM to manage identified presentation objects for presentation version control. Figure 6 shows pseudo code of how to create and manage an OSM tree. The first part is an OSM tree constructor that creates an initial OSM tree from an HTML page. The second part is an OSM tree adjustor that manages an established OSM tree upon receiving an access request.

Figure 6. Pseudo code of OSM tree creation and management.

```
//1. Constructor that creates an initiate OSM tree
 void OSMTree(HTML page) {
    createDOMTree(HTML page);
    traverse(DOMTree);
    for (each leaf node) {
       createMMO();
       createDefaultModalityNode();
       createDefaultFidelityNode();
       createPresentationObject();
    }
 }


//2. Upon receiving a presentation request
 void adjustOSMTree (request) {
    gatherContext(request);
```

```
    traverse(OSMTree);
    for (each MMO) {
       if (!satisfy(defaultModality, context) && !satisfy(anyExistingModality, context)) {
            createModalityNode(context);
            createFidelityNode(context);
            createPresentationObject(context);
       }
       if (!satisfy(defaultFidelity, context) && !satisfy(anyExistingFidelity, context)) {
          createFidelityNode(context);
          createPresentationObject(context);
       }
    }
}
```

To create an OSM tree, first of all, a DOM tree is generated by decomposing a formatted HTML page. Then the DOM tree is traversed, using any traversal algorithm such as pre-order or post-order, where each node is visited before or after any of its child nodes. Each leaf node in the DOM tree is replaced by a newly created MMO instance, with a default modality child node that in turn has a default fidelity child node. The default nodes are defined by the specifications in the corresponding HTML page. Meanwhile, a presentation node is created and stored into the *object repository* that can be accessed from the corresponding fidelity node. In other words, for an initiated OSM tree, each leaf node has defined a default presentation path comprising 4-tuple: <MMO, modality, fidelity, presentationObject>.

In Figure 4, each default link is represented by a solid directed line; each dynamically generated link is represented by a dotted directed line. For example, *Obj4* is by default represented as an image in the format of JPEG 1024x728. To serve different devices, *Obj4* can be adapted into different image formats such as BMP 24-bits or BMP 16-bits. To adapt to different user environments, *Obj4* can be adapted into other presentation types, such as text, audio, or video. Note that for simplicity reason, Figure 3 skips internal

nodes in the structure layer representing HTML tags.

Upon receiving an access request, the receiving contextual information has to be first gathered. Then the OSM tree is traversed in any order. For each MMO object, we check whether the default modality or existing modalities can satisfy the receiving contexts. If the answer is a "no," a new modality node will be created; then in turn a new fidelity node will be created. A new presentation object will also be created using adaptation tools, which will be discussed in detail in later sections. If the answer is a "yes," however, we need to first check whether its default fidelity or existing fidelities can satisfy the receiving contexts. If not, a new fidelity node will be created, together with a new presentation object. The dynamic content adaptation process will be discussed in the next section.

Dynamic Content Adaptation With Transcoding And Cache Management

If existing modality or fidelity formats cannot be found in the *object repository* for a presentation object, a *dynamic transcoding* process is invoked. As shown in Figure 4, the transcoded objects are cached and stored into *object repository* for future reuse. The process

may employ a set of transcoding apparatus dedicated for transforming different types of presentation objects into various modalities and the fidelities. We have experimented a variety of third-party transcoding tools. Among them, we found that the following four tools show satisfactory results for proprietary presentation transformation: VCDGear as a toolbox for video transformation, PictView as a command-line-driven image converter, and JafSoft (Chebotko, Lu et al. 2007) and Microsoft Reader (Microsoft) as text-to-audio (and vice versa) converters.

The design of our framework allows any third-part transcoding service to be easily plugged into the system, by adopting the strategy design pattern (Gamma, Helm et al. 1995). Any transcoding tools and services can be registered at the transcoder with specific functionalities. Upon receiving a transcoding request, the transcoder dynamically decides the most appropriate service and redirects the request to the selected service.

We enable two types of content adaptation, either static or dynamic. For static content adaptation, contents are transcoded and prepared before users' requests. Thus, content adaptation can only occur when applicable objects can be found in the *object repository*. No computational overhead is imposed in this manner because the actual transcoding process is performed offline. The problem of the static approach, however, is that the system has to predict all potential Web page users beforehand. For dynamic content adaptation, the system performs dynamic transcoding and caching if applicable objects cannot be found in the *object repository*. The advantage of this dynamic approach is the flexibility it offers: users can access any Web page and obtain adapted content upon requests. Its problem is that the performance of Web access may degrade due to additional transcoding overhead at run time. Users probably have to wait a significant amount of time when they access adapted Web content at the first time. Nevertheless, the performance will be enhanced the next time when the same user or other users access the same adapted Web content. The reason is that we have applied a caching mechanism presented by Kinno et al. (Kinno, Yukitomo et al. 2004), so that the transcoding overhead can be reduced. Note that instead of caching the entire adapted Web page, only the dynamically transcoded objects will be cached by our *cache management*. When dynamic content adaptation begins, the dynamically generated layout template and the associated objects will also be cached.

Because of the cache management, when users re-visit the same Web page, the access performance with dynamic content adaptation is almost the same as that without content adaptation. For PDA and mobile phone users, the performance of dynamic content adaptation is even better. This is because the size of the transcoded objects will be reduced, which results in tremendous saving of transmission bandwidth.

Page Layout Template Composition

The *layout template composition* phase is used to decide presentation styles (i.e., page layout templates) at run time. To enable fully personalized content adaptation, it is critical to allow end users to customize their own style sheets, to a certain level of granularity. To achieve this goal, we design an Extensible Stylesheet Language (XSL) style sheets-based strategy.

As shown in Figure 7, we have designed different page layout templates, for NBs, for PDAs, and for wireless phones. For each presentation type, each receiving device is associated with a proprietary style sheet. For example, since all three devices can present images and texts, their

corresponding style sheets for the two presentation types are the same. However, wireless phones may not be able to play video clips; PDAs may not intend to show all video clips unless users explicitly decide to do so. Therefore, as shown in Figure 7, the style sheet for NBs allows the source of a video clip to be embedded into the delivered content; while the style sheets for PDAs and phones only embed a link to the source of the video clip.

Figure 7. An example of XSL temples for NBs, PDAs, and phones.

Template	XSL for NB	XSL for PDA	XSL for phone
image	… <xsl:template match="**image**"> </xsl:template> …	… <xsl:template match="**image**"> </xsl:template> …	… <xsl:template match="**image**"> </xsl:template> …
Video	… <xsl:template match="**video**"> <embed src="**{@src}**" autostart="**false**" /> </xsl:template> …	… <xsl:template match="**video**"> **Video_Link** </xsl:template> …	… <xsl:template match="**video**"> **Video_Link** </xsl:template> ...
Text	**…** <xsl:template match="**text**"> <xsl:value-of select="**.**" /> </xsl:template> </xsl:stylesheet> …	… <xsl:template match="**text**"> <xsl:value-of select="**.**" /> </xsl:template> …	… <xsl:template match="**text**"> <xsl:value-of select="**.**" /> </xsl:template> …
Aggregated XSL file	<?xml version="1.0" encoding="UTF-8" ?> - <xsl:stylesheet version="**1.0**" xmlns:xsl=http://www.w3.org/1999/XSL/Transform/> <xsl:import href="**body.xsl**" /> <xsl:import href="**p.xsl**" /> <xsl:import href="**table_nb.xsl**" /> <xsl:import href="**tr_nb.xsl**" /> <xsl:import href="**td_nb.xsl**" /> <xsl:import href="**text.xsl**" /> <xsl:import href="**image_play.xsl**" /> <xsl:import href="**audio_play.xsl**" /> <xsl:import href="**video_play.xsl**" /> - <xsl:template	<?xml version="1.0" encoding="UTF-8" ?> <xsl:stylesheet version="1.0" xmlns:xsl=http://www.w3.org/1999/XSL/Transform/> <xsl:import href="**body.xsl**" /> <xsl:import href="**p.xsl**" /> <xsl:import href="**table_pda.xsl**" /> <xsl:import href="**tr_pda.xsl**" /> <xsl:import href="**td_pda.xsl**" /> <xsl:import href="**text.xsl**" /> <xsl:import href="**image_play.xsl**" /> <xsl:import href="**audio_link.xsl**" /> <xsl:import href="**video_link.xsl**" /> - <xsl:template match="**html**"/> - <html> <xsl:apply-templates. />	<?xml version="1.0" encoding="UTF-8" ?> - <xsl:stylesheet version="**1.0**" xmlns:xsl=http://www.w3.org/1999/XSL/Transform/> <xsl:import href="**body.xsl**" /> <xsl:import href="**p.xsl**" /> <xsl:import href="**table_phone.xsl**" > <xsl:import href="**tr_phone.xsl**" /> <xsl:import href="**td_phone.xsl**" /> <xsl:import href="**text.xsl**" /> <xsl:import href="**image_play.xsl**" /> <xsl:import href="**audio_link.xsl**" /> <xsl:import href="**video_link.xsl**" /> - <xsl:template match="**html**"/> - <html>

match="**html**"/> - <html> <xsl:apply-templates. /> </html> <xsl:template> </xsl:stylesheet>	</html> <xsl:template> </xsl:stylesheet>	<xsl:apply-templates. /> </html> <xsl:template> </xsl:stylesheet>

As shown in Figure 7, each receiving device is also associated with some other utility style sheets. For example, in order to present a table, NBs are associated with a "table_nb.xsl"; PDAs are associated with a "table_pda.xsl"; phones are associated with a "table_phone.xsl." We design a set of utility style sheet templates for all receiving device types. Some other examples are illustrated in Figure 7, such as "tr" (row), "td" (column), "image_play," "audio_link," and "video_link."

These individual style sheets are configurable and re-configurable by different end users for different receiving devices. At the phase of *layout template composition*, as shown in Figure 6, an aggregated XSL style sheet file is created, by importing all related style sheets serving various purposes (for both presentation types and supporting utilities).

Our strategy of separation of concerns allows configuration and re-configuration of personalizable/customizable presentation styles at a fine-grained granularity. Our dynamic aggregation strategy ensures a format management solution, with high flexibility and extensibility.

As mentioned in *Web page decomposition* phase in section 5.1, the inter-object relationships (pre-order or post-order sequences) are used to restore an object's presentation sequence and generate an adapted Web page. We adopt post-order sequence to describe parallel order of objects. For PC and NB, one object can be displayed beside or next to another object in terms of their spatial relationships, or one object can be rendered concurrently with another object in terms of their temporal relationships. In contrast, pre-order sequence is used to describe sequential order of objects. For smaller screen size of PDAs and mobile phones, one object can be displayed above or below another object in terms of their spatial relationships, or one object can be rendered before or after another object in terms of their temporal relationships.

IMPLEMENTATION DETAILS

In this section, we discuss some of our implantation details, focusing on the aspects of context definition and design pattern usages. In our research, we adopt OWL to build our ontology, as OWL is based on DLs and is natural for us to define receiving contexts and requirements.

Rule-Based Adaptation Management Using Jess

We have adopted JESS (JESS) (Jess Expert System Shell), a rule engine for the Java platform, to design and implement content adaptation rules. Being a superset of CLIPS programming language, JESS provides a rule-based paradigm suitable for automating an expert system. In contrast to a procedural programming language where a single program is activated at one time, the JESS' declarative paradigm allows us to apply a collection of rules to a collection of facts based on pattern matching. In addition, JESS is Java-based and conforms to our implementation environment.

Examples of rules over conditions are listed in Figure 8. The JESS rules define the template of an object as a 3-tuple (object-ID,

modality, fidelity), and then define the template of a situated environment as a 4-tuple (pserson-ID, condition, network, device). Figure 8 defines 5 rules over different conditions. Take rule 1 as an example. The receiving environment is normal; thus, rule 1 requires no special action except printing out a message at DOS prompt. Take rule 5 as another example. The receiving environment is noisy and it allows video and audio objects. Thus, rule 5 requires checking object identifier, verifying the modality of the presentation objects in the format of video or audio; and turning the fidelity of the presentation objects to be loud.

Figure 8. Examples of content adaptation rules written in JESS.

```
;; Facts for describing object
  (deftemplate Object "define object modality and fidelity"
                (slot object-ID)
                (slot Modality)
                 (slot Fidelity))
;; Facts for describing users situated environment
(deftemplate Situated-Env " define users situated environment "
                (slot Person-ID)
                (slot Condition)
                (slot Network)
                 (slot Device))
;;    Condition Rules
(defrule Condition-r1    (Situated-Env (Condition normal))
                  =>
                  (printout t "no adaptation while at normal condition" crlf)))
(defrule Condition-r2    ((Object ((ObjID ?objID) (Modality ?mod& image | video)))
                  (Situated-Env (Condition?cond & dark | driving)))
                  =>
                  (assert (Object (ObjID ?objID) (Modality ?mod) (Fidelity blank))))
(defrule Condition-r3    ((Object ((ObjID ?objID) (Modality ?mod& image | video | text)))
                  (Situated-Env (Condition blurred )))
                  =>
                  (assert (Object (ObjID ?objID) (Modality ?mod) (Fidelity bright))))
(defrule Condition-r4    ((Object ((ObjID ?objID) (Modality ?mod& video | audio)))
                  (Situated-Env (Condition?cond & quiet | meeting)))
                  =>
                  (assert (Object (ObjID ?objID) (Modality ?mod) (Fidelity mute))))
(defrule Condition-r5    ((Object ((ObjID ?objID) (Modality ?mod& video | audio)))
                  (Situated-Env (Condition noisy)))
                  =>
                  (assert (Object (ObjID ?objID) (Modality ?mod) (Fidelity loud))))
```

Design Pattern Usages

In the implementation of our content adaptation planner, we adopted three design patterns (Gamma, Helm et al. 1995) to enhance design and code reusability, flexibility, and extensibility. The four patterns are: lazy initiation, Model-Viewer-Controller (MVC), mediator, and strategy.

In the previous section, we discussed our strategy of adopting the lazy initiation design pattern in constructing an OSM tree. In the initialization phase, only the default presentation path is detected and created. All

other possible presentation paths (including modality, fidelity, and presentation object) are created at an on-demand basis.

In order to offer dynamic content adaptation to provide various presentation formats, we adopted the Model-View-Controller (MVC) design pattern. The MVC pattern has been known to facilitate maintenance by separating code for modeling the application domain from its GUI code. Figure 9 illustrates how we use the MVC pattern. As discussed earlier, we use our OSM tree to enable and manage multiple presentation formats of an HTML page. Inside of an OSM tree, each MMO represents a model for a specific presentation object; its associated adaptation rules represent a controller; any of its presentation paths (either the default path or dynamically generated paths) represents a view (presentation) of the modeled presentation object. Note that the multiplicity relationship between MMO and adaptation rule is a one-to-many relationship; the multiplicity relationship between MMO and presentation path is also a one-to-many relationship. This means that under different adaptation rules, a presentation object may show different presentation formats.

Figure 9. Model-View-Controller design pattern adopted.
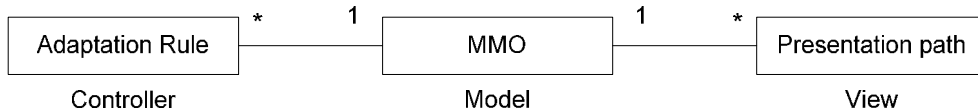
When we implemented the dynamic content adaptation engine, we adopted the mediator pattern and the strategy pattern, as shown in Figure 10. Both of them are in the category of behavioral patterns. The mediator pattern intends to offer a unified interface to a set of interfaces in a subsystem; the strategy pattern intends to provide an agile technique so that algorithms can be selected on-the-fly at runtime.

Figure 10. Mediator and Strategy patterns adopted.
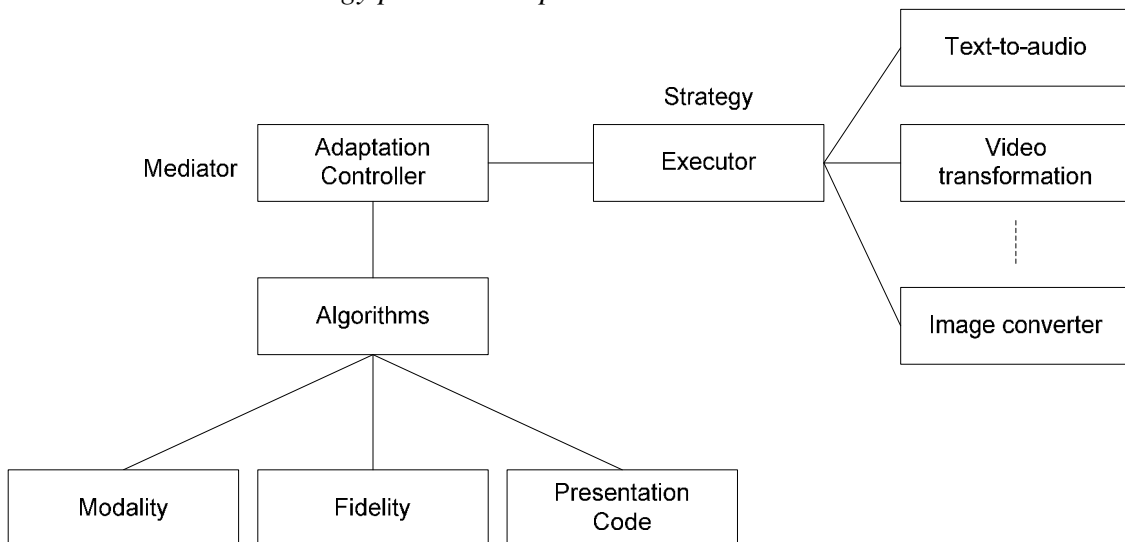
As mentioned earlier, without losing generality, in this research we consider four modalities: text, image, audio, and video. Based on various receiving contexts, a presentation object in one modality may need to be transformed into another modality. For example, a text message can be transformed into an audio clip for a user to listen instead of reading, assuming the user is driving. The maximum number of categories on content modality adaptation can be calculated as: $P_4^2 = 12$. In addition, even staying with the same modality, a presentation object can change its fidelity. For example, an image object can be adapted from a high resolution into a lower resolution.

In order to enable one interface to serve all possible content adaptation requests, we adopted the mediator design pattern. As shown in Figure 10, an adaptation controller provides a unified interface for any content adaptation requests. Based on its carried algorithms and parsers, the adaptation controller examines an incoming request and analyzes its details such as its goal, input parameters, output parameters, context, and so on. Then the adaptation controller directs the request to proper executor.

To design the executor, we adopted the strategy design pattern. As discussed in Section 4.3, any transcoding tools and services can be registered at the executor with specific functionalities. Upon receiving an adaptation request, the executor dynamically decides the most appropriate service and redirects the request to the selected service.

Lazy Initialization Of OSM Trees

An OSM tree is dynamically created and maintained. A *lazy instantiation* design pattern (Gamma, Helm et al. 1995) is used here. At the initial construction, only the default presentation path is created. Other presentat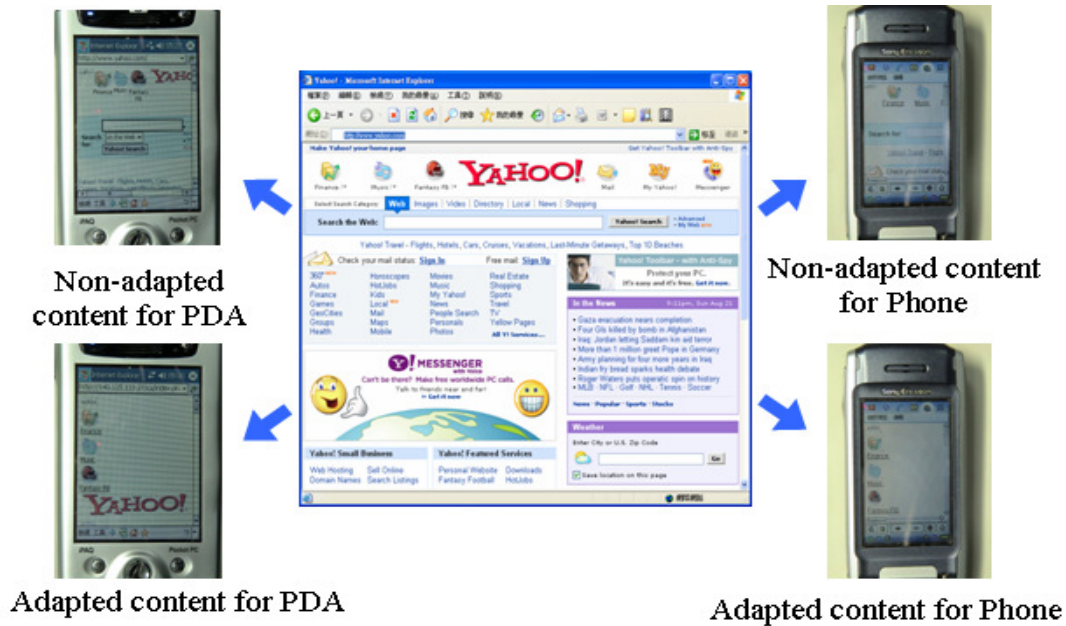ion paths are created based on an on-demand basis. All dynamically created MMO objects, modality objects, fidelity objects, and presentation objects are stored in the *object repository* for later reuse. The OSM tree is also stored in the *object repository*. Upon the first access, an OSM tree is created for a specific Web page. We realize their linkage through a hyperlink. The hyperlink concept was originally created to represent a link embedded in an HTML page to redirect access to another URL. We extend the hyperlink concept to realize association of an OSM tree to its HTML page. In other words, a hyperlink pointing to an OSM tree is embedded into each Web page to manage its dynamic content adaptation and presentation in its entire life cycle.

EXPERIMENTS AND EVALUATIONS

We have implemented a content adaptation planner. Figure 11 illustrates some execution results of our content adaptation planner through a comparison of Web browsing between conventional and handheld devices. The middle of Figure 11 shows the Yahoo home page (http://www.yahoo.com/) on a desktop browser. The left-hand side of Figure 8 shows how this page is shown on a PDA screen without content adaptation (upper-side) and with content adaptation (lower-side). The right-hand side of Figure 11 shows how the same page is shown on a wireless phone screen without content adaptation (upper-side) and with content adaptation (lower-side). As shown in Figure 11, without content adaptation, users of PDAs and wireless phones have to move scroll bars left and right, up and down in order to view the whole Web page. In contrast, our content adaptation obviously provides better content presentation to the users by transforming the Web page into a column-wise presentation, so that users only need to move scroll bars in one direction

(up and down) instead of in two directions.

Figure 11. Snapshots of adapted Web page for NB, PDA, and Phone.

Non-adapted content for PDA

Adapted content for PDA

Non-adapted content for Phone

Adapted content for Phone

Measurement And Comparison Of Access Time

We have designed and conducted a set of experiments to measure and compare the access time of Web content with and without our context-driven content adaptation planner. The goal is to evaluate whether adopting our content adaptation planner could enhance mobile Web accessibility for users with handheld devices. The detailed experimental set up is shown in Figure 12. Yahoo (http://www.yahoo.com) was chosen as the content provider and test bed. The reasons why we chose this commonly accessed Web site are as follows. First, Yahoo's site is well known and its overall structure (i.e., its division of sections) is stable. Therefore, our tests can be easily repeated by third parties, and the test results are more trustworthy. Second, Yahoo's site comprises a comprehensive set of sections, each containing significantly different content types with proprietary static attributes and dynamic features. Therefore, our experiments and results will be more useful and convincing.

This experiment of content adaptation (CA in short) was designed to measure the content access time for a user to browse different types (i.e., sections or categories) of Yahoo Web pages. Without losing generality, we selected five sections of Yahoo pages, namely *Home*, *News*, *Astrology*, *Auto*, and *Movie*. Each section exhibits some specific features. For example, the *Home* section maintains a relatively stable structure; the *News* section is under frequent updates so that its structure is volatile; the *Astrology*, *Auto*, and *Movie* sections contain a lot of images, photos, audio, and video clips. In short, we used Yahoo's diversified sections as the test bed to evaluate the effectiveness and efficiency of our content adaptation planner.

Figure 12. Experimental set up.
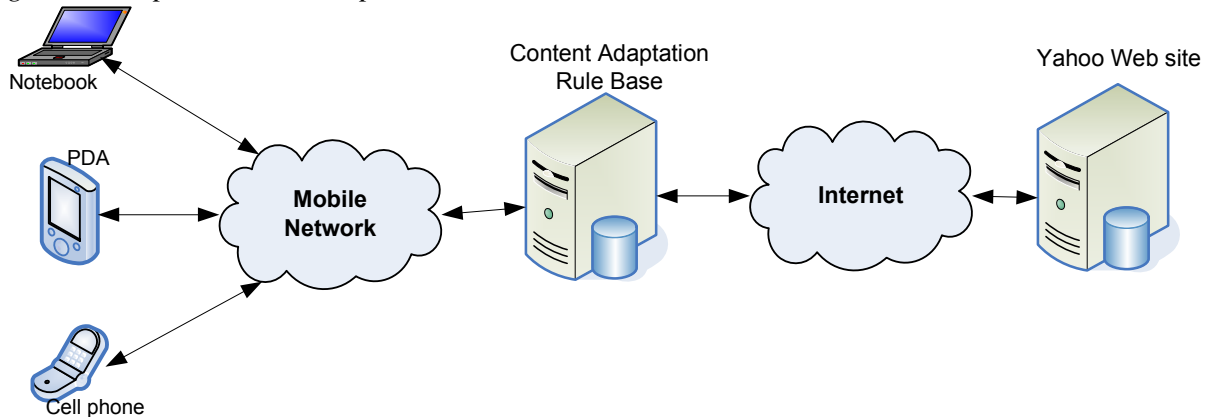
As shown in Figure 12, our content adaptation planner is implemented on an IBM server machine equipped with an Intel 1.0GHz CPU and a 1GB memory. The receiving devices we tested include a Compaq 3870 PDA with Microsoft Windows C.E. 3.0 PPC, a Sony Eriksson P900 smart phone with Symbian, and a notebook (NB) with Microsoft Windows XP installed. The connections between the receiving devices and the content adaptation planner are via a wireless network; the connection between the content adaptation planner and the Yahoo Web site is through the wired Internet. The contrast tests without using content adaptation planner are based on the same environmental settings.

The access time is measured for three different devices (NB, PDA, and phone) under three different conditions (NCA, CA_1, and CA_2). The experiment results are summarized in Table 1, 2, and 3. Various kinds of access time measured and denoted in this section are defined as follows:

- NCA: The access time for a user's visiting a Web page without content adaptation, which is the time interval between sending a request and receiving a reply.
- CA_1: The access time for a user's first visiting a Web page with content adaptation.
- CA_2: The access time for a user's second visiting a Web page with content adaptation.
- pre-CA process: The process before content adaptation, including Web page decomposition, trans-coding, object repository construction, and context acquisition (please refer to Figure 1 for content adaptation process).
- pre_CA%: The percentage of pre-CA processing time over the total access time.

When a user first visits a Web page, the content adaptation process has to go through three steps: first is the pre-CA process; then the adaptation planner decides an adaptation strategy; afterwards the adaptation engine executes the adaptation accordingly.

Table 1. First time visiting a Web page with content adaptation.

Yahoo page	device	pre-CA process (sec.)	adaptation planner (sec.)	adaptation engine (sec.)	total access time (sec.)	pre-CA%
home	NB	10.9630	0.2723	0.7928	12.0281	91.14%
news	NB	14.4520	0.3727	1.0151	15.8398	91.24%
astrology	NB	6.5877	0.2138	0.5842	7.3857	89.20%
auto	NB	6.8166	0.1587	0.4749	7.4502	91.50%
movie	NB	12.1584	0.3739	0.7615	13.2938	91.46%
home	PDA	10.9630	0.7561	7.0414	18.7605	58.44%
news	PDA	14.4520	0.8510	10.3604	25.6634	56.31%
astrology	PDA	6.5877	0.2195	7.8263	14.6335	45.02%
auto	PDA	6.8166	0.1617	6.7837	13.7620	49.53%
movie	PDA	12.1584	0.8858	10.3503	23.3945	51.97%
home	phone	10.9630	1.7266	21.8300	34.5196	31.76%
news	phone	14.4520	0.8473	26.7388	42.0381	34.38%
astrology	phone	6.5877	0.2185	15.7217	22.5279	29.24%
auto	phone	6.8166	0.6791	10.4697	17.9654	37.94%
movie	phone	12.1584	1.7736	22.0621	35.9941	33.78%

Table 1 shows the performance measurement of a user's first-time visit of a Web page with content adaptation. The total access time is the sum of the execution time from three parts: pre-CA process, adaptation planner, and adaptation engine. For each device's accessing one of the five Yahoo sections, we recorded the processing time (in seconds) of its three individual steps and the total access time, as well as the percentage of the pre-CA processing time over the total access time. As shown in Table 1, the pre_CA% for NB is significantly higher than that for PDA and for phone. This observation reveals that the bottleneck of the content adaptation process for NB is at the pre-CA process (around 91%).

Table 2 shows the performance measurement of the second-time visiting of the same Web page with content adaptation. When a user visits the same Web page for the second time, the content adaptation process can skip the pre-CA process so the total access time will be reduced. For each device's accessing one of the five Yahoo sections, we recorded the processing time (in seconds) of its two individual steps and the total access time, as well as the percentage of the processing time of the adaptation engine over the total access time. As shown in Table 2, the processing time of the adaptation engine is the bottleneck of the entire content adaptation for all three kinds of devices, especially for devices with less computational power (PDA and cell phone).

Table 2. Second-time visiting the same Web page with content adaptation.

Yahoo page	device	adaptation planner (sec.)	adaptation engine (sec.)	total access time (sec.)	adaptation engine%
home	NB	0.2723	0.7928	1.0651	74.43%
news	NB	0.3727	1.0151	1.3878	73.14%
astrology	NB	0.2138	0.5842	0.7980	73.21%
auto	NB	0.1587	0.4749	0.6336	74.95%
movie	NB	0.3739	0.7615	1.1354	67.07%
home	PDA	0.7561	7.0414	7.7975	90.30%
news	PDA	0.8510	10.3604	11.2114	92.41%
astrology	PDA	0.2195	7.8263	8.0458	97.27%
auto	PDA	0.1617	6.7837	6.9454	97.67%
movie	PDA	0.8858	10.3503	11.2361	92.12%
home	phone	1.7266	21.8300	23.5566	92.67%
news	phone	0.8473	26.7388	27.5861	96.93%
astrology	phone	0.2185	15.7217	15.9402	98.63%
auto	phone	0.6791	10.4697	11.1488	93.91%
movie	phone	1.7736	22.0621	23.8357	92.56%

Definition 8. $CA\_1_{overhead}$ denotes the overhead due to content adaptation at user's first-time access, $CA\_1_{overhead}$ is the overhead due to content adaptation at user's second-time access:

$$CA\_1_{overhead} = \frac{(access\_time_{CA\_1} - access\_time_{NCA})}{access\_time_{NCA}}$$

$$CA\_2_{overhead} = \frac{(access\_time_{CA\_2} - access\_time_{NCA})}{access\_time_{NCA}}$$

The higher the index, the more overhead it represents. If the value of an index is negative, it means that the content adaptation results in time saving instead of overhead. To better illustrate the overhead comparison, we further use a line chart in Figure 13 to represent the data shown in Table 3.

As shown in Figure 13 and Table 3, content adaptation does bring overhead to NB at the first time of content access, as indicated by the column $CA\_1_{overhead}$. However, when accessing the same content at the second time, the values of the corresponding $CA\_2_{overhead}$ column become negative, meaning that it takes less time to access the same content. This is because the adapted content has been processed and cached by our adaptation planner. The time for accessing adapted content is significantly less than that for accessing the original content due to the significant reduction in content size.

Table 3. Overhead of the first-time visit and second-time visit on the same Web page.

Yahoo page	device	access time for NCA (sec.)	access time for CA_1 (sec.)	access time for CA_2 (sec.)	CA_1$_{overhead}$	CA_2$_{overhead}$
home	NB	1.6470	12.0281	1.0651	630.30%	-35.33%
news	NB	3.6064	15.8398	1.3878	339.21%	-61.52%
astrology	NB	2.2178	7.3857	0.7980	233.02%	-64.02%
auto	NB	2.1850	7.4502	0.6336	240.97%	-71.00%
movie	NB	2.5448	13.2938	1.1354	422.39%	-55.38%
home	PDA	10.4060	18.7605	7.7975	80.29%	-25.07%
news	PDA	28.8180	25.6634	11.2114	-10.95%	-61.10%
astrology	PDA	12.4760	14.6335	8.0458	17.29%	-35.51%
auto	PDA	16.4620	13.7620	6.9454	-16.40%	-57.81%
movie	PDA	21.1840	23.3945	11.2361	10.43%	-46.96%
home	phone	43.8100	34.5196	23.5566	-21.21%	-46.23%
news	phone	119.7560	42.0381	27.5861	-64.90%	-76.96%
astrology	phone	68.5200	22.5279	15.9402	-67.12%	-76.74%
auto	phone	83.9000	17.9654	11.1488	-78.59%	-86.71%
movie	phone	120.3380	35.9941	23.8357	-70.09%	-80.19%

Figure 13. CA_1 overhead vs CA_2 overhead.
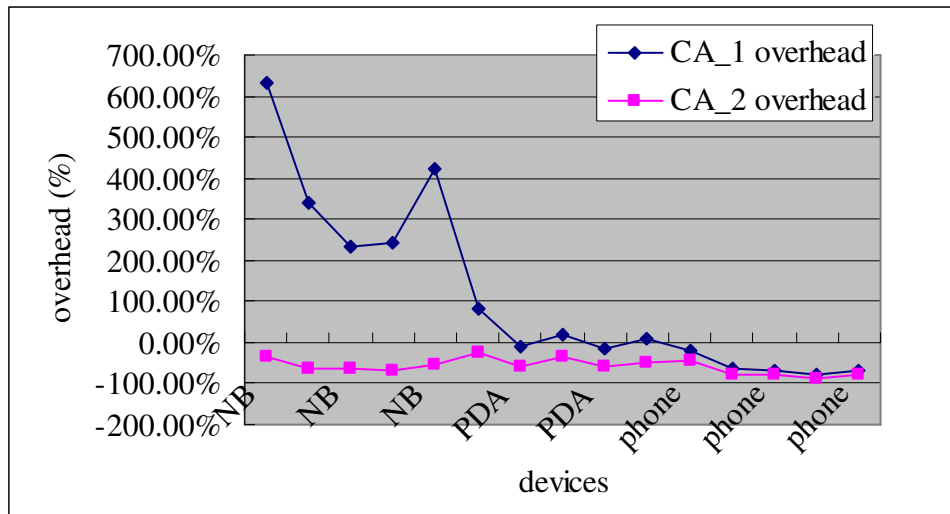
It should be noted that such phenomenon of saving content access time due to content adaptation becomes more significant when users are using portable devices. As shown in Table 3, a phone's content access time is reduced dramatically with content adaptation, no matter whether it is the first-time or the

second-time visit. Similar results are observed from PDAs. This is mainly because the content adaptation process has utilized our adaptation planners to change the content's modality and fidelity based on the users' situated environment, which results in a tremendous saving of bandwidth. In other words, our experimental results show that the computational overhead due to the context-driven content adaptation rules can be moderately decreased through the reduction of the size of adapted content.

Qualitative Evaluation Of Users' Satisfactions

Under the same experimental set up as described in Figure 11, we further designed a qualitative evaluation process to test users' satisfaction levels on the quality of adapted contents using our context-driven content adaptation planner. The motivation is that we realize content adaptation might result in degradation of content quality, which may affect people's comprehension of the adapted content. The reasons of quality degrade might include: reading difficulties due to either a device's screen size or improper content resizing or content reorganization, degrade in fidelity due to either devices' computing and bandwidth limitation or improper trans-coding, and so on. Thus, we designed four evaluation questions to evaluate users' satisfactions with the quality of adapted images, video clips, texts, and the experience of page navigation.

1. Are testers satisfied with the quality of adapted image?
2. Are testers satisfied with the quality of adapted video clip?
3. Are testers satisfied with the quality of text browsing on the adapted page?
4. Are testers satisfied with the quality of page navigation on the adapted page?

43 randomly selected student testers participate in this evaluation. Each of them was asked to use PDAs and mobile phones to access the aforementioned five sections of Yahoo pages (*home*, *news*, *astrology*, *auto*, and *movie*) after content adaptation, and then record their answers to the four questions above. Based on their responses, we summarize our findings in Table 4. When testers use PDAs, the *home* section provides the best results of content adaptation because it has the most stable Web page structure. 83% of the testers are satisfied with the adapted image on the *home* section. Regarding the experience of text browsing and page navigation on the *home* section, we found that most of the testers are satisfied with the adaptation. (96% and 94% of the testers have good experience of text browsing and page navigation on the adapted page, respectively, which are the two highest degrees of satisfaction in this evaluation experiment).

However, when the testers use mobile phones, they showed a lower degree of satisfaction with the adapted content due to the smaller screen size and less computing capability, especially with adapted images and adapted video clips on the *news* section due to its nature of frequent change. (On the *news* section, only 47% and 43% of the testers are satisfied with adapted *images* and *video clips*, respectively, which are the two lowest degrees of satisfaction in this evaluation experiment).

The testers also expressed their strong demands of content adaptation when they browsed those image-oriented Web pages such as the sections of *astrology*, *auto*, and *movie*. They indicated that a flexible and readable content presentation will be one of their major concerns when they use handheld devices for mobile Internet surfing.

Table 4. Evaluation results of testers' satisfactions with the quality of adapted content on five different Yahoo Web sections.

Yahoo page	Content types	Degree of satisfaction with PDA (%)	Degree of satisfaction with phone (%)
home	image	83%	67%
	video clip	80%	56%
	text browsing	96%	87%
	page navigation	94%	85%
news	image	68%	47%
	video clip	63%	43%
	text browsing	75%	78%
	page navigation	71%	72%
astrology	image	73%	56%
	video clip	68%	53%
	text browsing	90%	80%
	page navigation	92%	73%
auto	image	79%	57%
	video clip	75%	51%
	text browsing	87%	82%
	page navigation	89%	74%
movie	image	80%	60%
	video clip	81%	55%
	text browsing	86%	81%
	page navigation	85%	77%

Further Discussions

Although content adaptation can provide better content presentation based on people's contextual environment, it may also cause significant computational overhead. Therefore, one might argue that there is no need to perform content adaptation such as turning off the video while a person is driving. However, one major reason why such adaptation is valuable is that it could largely improve transmission performance, especially when transmitting rich media over wireless networks. For example, removing unnecessary video may largely save transmission time.

Although our experiments test access time for one user and prove that content adaptation overhead can be reduced after the user's first access, the observation can be extended to multi-user situation. The reason is that our content adaptation is conducted at the content adaptation planner acting as a proxy serving multiple users. After any user

accesses a page, its adapted content will be cached, and all later users will benefit as long as they have similar receiving conditions. Thus, the more popular a Web page is, the more users will benefit from our technique.

In addition, not every handheld device can play all types of media. For example, a non-multimedia cell phone cannot play video clips. Hence, by removing such a multimedia component if it is un-playable on receiving devices, content adaptation could save unnecessary network bandwidth. Another alternative is to perform transcoding to downgrade such a multimedia component into a playable format, e.g., downgrade a video clip to static images. Nevertheless, there is always a tradeoff between quality and performance, and this is one of the motivations why we design a dynamic content adaptation planner.

Furthermore, in a mobile computing environment, context is typically a generalized concept and requires specialization based on specific application logic (Julien & Roman 2006). Our study here provides a guidance and prototype for people to build domain-specific content adaptation and delivery services.

CONCLUSIONS

In this paper, we present a context-driven content adaptation planner concerning users' situated contextual environments. Our experimental results show that content adaptation can dramatically reduce the access time to Web content, especially when users are using handheld devices over the mobile Internet.

Our content adaptation planner may also benefit people who are with disability such as deaf and blind. For people who are deaf, they are in the same condition as people who are sitting in a meeting because both of them will not hear the audio. For people who are blind,

they are in the same condition as people who are driving because both of them will not read the content. For those who suffer from weak vision, they are in the same condition as people who are in a blurred environment (e.g., due to a sunny or gloomy weather); the content should be consequently enlarged or the background color should be turned brighter. For people who suffer from weak hearing, they are in the same condition as people who are in a noisy environment (such as in a marketplace); then the audio volume should be turned louder.

The main contribution of this paper is the development of a context-driven content adaptation planner to enhance the Web page content accessibility using handheld devices over the mobile Internet. We apply description logics (DLs) to formally define context profiles and requirements and automate content adaptation decision. Experimental results show that the computation overhead caused by the context-driven content adaptation planner can be moderately decreased through multiple accesses and through the reduction of the size of adapted content. The saving of transmission bandwidth becomes more significant when users are using handheld devices over the mobile Internet.

Our presented content adaptation planner provides some solutions to the four research challenges we identified at the beginning of this research project. For the challenge of *detection and representation of mobile user contexts*, we present a multi-dimensional formal model (including device, environment, status, and end user profile) that is then used to guide content adaptation. For the challenge of *adaptation rule design*, we present a JESS-enabled rule engine with an incrementally constructed rule base. For the challenge of *content adaptation management*, we present the design and implementation of content adaptation planner that balances the

tradeoffs of various concerns. A formal OSM model is proposed as the fundamental basis for content adaptation management with high reusability and adaptability. Algorithms and detailed designs are presented for guiding automatic and dynamic content adaptation. For the challenge of *format (e.g., style sheet) generation*, we present our fine-grained style sheet management solution to enable configurable and re-configurable presentation style control.

Nevertheless, we also found several barriers existing in our current adaptation techniques, such as lacking the capability of processing script languages (e.g., JavaScript and VBScript), lacking the mechanism of processing image without alternative descriptions and voice recognition.

In addition, we found that many Web pages have similar layout structures, even though their contents are significantly different. Some similar and rarely changed portions (e.g., header fragment, navigation fragment, and advertisement fragment) occupy significant amount of storage space and consume computing resources to process. To further speed up the decomposition process and reduce required storage space, we are currently exploring a dedicated fragment detection method for automatically detecting similar fragments by identifying the coding patterns (Gamma, Helm et al. 1995) among Web pages.

Furthermore, currently we allow individual end users to configure and re-configure their dedicated style sheets. However, this extremely fine-grained solution is impractical since the cost of style sheet management is propositional to the number of end users. It may become too costly to maintain a large amount of style sheets. A more feasible solution is to attach a spread sheet with a group of users with common features. However, how to control the granularity of user groups remains

challenging and will be investigated in our future research.

ACKNOWLEDGEMENTS

REFERENCES

"PictView." from http://www.pictview.com/pvw.htm.

"VCDGear." from http://www.vcdgear.com.

Baader,F., Calvanese,D., McGuinness, D.L., Nardi, D., & Patel-Schneider, P.F. (2003). *The Description Logic Handbook: Theory, Implementation, Applications*. Cambridge, UK, Cambridge University Press.

Bellavista, P., Corradi, A.,Montanari, R. ,& Stefanelli, C. (2003). Context-Aware Middleware for Resource Management in the Wireless Internet. *IEEE Transactions on Software Engineering*. 29(12): 1086-1099.

Berhe,G., Brunie, L., & Pierson, J.M. (2004). Modeling Service-Based Multimedia Content Adaptation in Pervasive Computing. *the First Conference on Computing Frontiers on Computing Frontiers*, ACM Press, 60-69.

Bose, R., & Frew, J. (2005). Lineage retrieval for scientific data processing: a survey. *ACM Comput*. Surv. 37(1): 1-28.

Cabri,G., Leonardi, L., Mamei, M., & Zambonelli, F. (2003). Location-Dependent Services for Mobile Users. *IEEE*

Transactions on Systems, Man and Cybernetics, Part A. 33(6): 667-681.

Chebotko, A., S, Lu., & Fotouhi, F. (2007). Semantics preserving SPARQL-to-SQL query translation. *Data and Knowledge Engineering*, Submitted.

Chen, L.Q., Xie, X., Fan, X., Ma, W.Y., Zhang, H.J., & Zhou, H.Q. (2002b). *A Visual Attention Model for Adapting Images on Small Displays*. Technical report MSR-TR-2002-125, Microsoft Research.

Chen, L.Q., Xie, X., Ma, W.Y., Zhang, H.J., Zhou, H.Q. & Feng, H.Q. (2002a). *DRESS: A Slicing Tree Based Web Representation for Various Display Sizes*. Technical report MSR-TR-2002-126, Microsoft Research.

Dey, A.K., & Abowd, G.D. (1999). *Toward A Better Understanding of Context and Context-Awareness*. GVU Technical Report GIT-GVU-99-22, College of Computing, Georgia Institute of Technology.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison Wesley, Boston, MA, USA.

He, J., Gao, T., Hao, W., Yen , I.-L., & Bastani, F. (2007). A Flexible Content Adaptation System Using a Rule-Based Approach. *IEEE Transactions on Knowledge and Data Engineering*. 19(1): 127-140.

Hua, Z., Xie,X., Liu,H., Lu, H., & Ma, W.-Y. (2006). Design and Performance Studies of an Adaptive Scheme for Serving Dynamic Web Content in a Mobile Computing Environment. *IEEE Transactions on Mobile Computing*. 5(12): 1650-1662.

JESS. "Jess Expert System Shell." from http://www.jessrules.com/.

Julien, C., & Roman, G.-C. (2006). EgoSpaces: Facilitating Rapid Development of Context-Aware Mobile Applications. *IEEE Transactions on Software Engineering*. 32(5): 281-298.

Khedr, M., & Karmouch, A. (2004). Negotiating Context Information in Context-Aware Systems. *IEEE Intelligent Systems.* 19(6): 21-29.

Kinno, A., Yukitomo, H., & Nakayama, T. (2004). An Efficient Caching Mechanism for XML Content Adaptation. *the 10th International Multimedia Modeling Conference* (MMM'04), Jan., IEEE Press, 308-315.

Krause, A., Smailagic, A., & Siewiorek, D.P. (2006). Context-Aware Mobile Computing: Learning Context- Dependent Personal Preferences from A Wearable Sensor Array. *IEEE Transactions on Mobile Computing*. 5(2): 113-127.

Kurz, B., Popescu, I., & Gallacher, S. (2004). FACADE - A Framework for Context-Aware Content Adaptation and Delivery. *Second Annual Conference on Communication Networks and Services Research*, IEEE CS Press, 46-55.

Lee, Y.W., Chandranmenon, G., & Miller, S.C. (2003). *GAMMA: A Content Adaptation Server for Wireless Multimedia Applications*. Bell-Labs, Technical Report.

Lei, Z. & Georganas, N.D. (2001). Context-based Media Adaptation in Pervasive Computing. *Canadian Conference on Electrical and Computer Engineering*,

May,

Lemlouma, T., & Layaida, N. (2001). The Negotiation of Multimedia Content Services in Heterogeneous Environments. *The 8th International Conference on Multimedia Modeling (MMM)*, Amsterdam, The Netherlands, Nov. 5-7, 187-206.

Lemlouma, T., & Layaida, N. (2004). Context-Aware Adaptation for Mobile Devices. *2004 IEEE International Conference on Mobile Data Management*, IEEE CS Press, 106-111.

Lum, W.Y., & Lau, F.C.M. (2002). A Context-Aware Decision Engine for Content Adaptation. *IEEE Pervasive Computing*. 1(3): 41-49.

Microsoft. "*Microsoft Reader*." from http://www.microsoft.com.

Mohan, R., Smith, J.R., & Li, C.S. (1999). Adapting Multimedia Internet Content for Universal Access. *IEEE Transactions on Multimedia*. 1(1): 104-114.

Mostefaoui, S.K., & Hirsbrunner, B. (2004). Context Aware Service Provisioning. *the IEEE International Conference on Pervasive Services (ICPS)*, Beirut, Lebanon, Jul. 19-23, 71-80.

Mostefaoui, S.K., Tafat-Bouzid, A., & Hirsbrunner, B. (2003). Using Context Information for Service Discovery and Composition. *5th International Conference on Information Integration and Web-based Applications and Services (iiWAS)*, Jakarta, Indonesia, Sep. 15-17, 129-138.

Mukherjee, D., Delfosse, E., Kim, J.-G., & Wang, Y. (2005). Optimal Adaptation Decision-Taking for Terminal and Network

Quality-of-Service. *IEEE Transactions on Multimedia*. 7(3): 454-462.

Nam, J., Ro, Y.M., Huh, Y., & Kim, M. (2005). Visual Content Adaptation According to User Perception Characteristics. *IEEE Transactions on Multimedia*. 7(3): 435-445.

Pashtan, A., Kollipara, S., & Pearce, M. (2003). Adapting Content for Wireless Web Service. *IEEE Internet Computing*. 7(5): 79-85.

Phan, T., Zorpas, G., & Bagrodia, R. (2002). An Extensible and Scalable Content Adaptation Pipeline Architecture to Support Heterogeneous Clients. *the 22nd International Conference on Distributed Computing Systems*, IEEE CS Press, 507-516.

Ramaswamy, L., Iyengar, A., Liu, L., & Douglis, F. (2005). Automatic Fragment Detection in Dynamic Web Pages and Its Impact on Caching. *IEEE Transactions on Knowledge and Data engineering*. 17(6): 859-874.

Roman, M., Hess, C.K., Cerqueira, R., Ranganathan, A., Campbell, R.H., & Nahrstedt, K. (2002). Gaia: A Middleware Infrastructure to Enable Active Spaces. *IEEE Pervasive Computing*. 1(4): 74-83.

M. Satyanarayanan, (2004). The Many Faces of Adaptation. *IEEE Pervasive Computing*. 3(3): 4-5.

Schilit, B.N., Adams, N.I., & Want, R. (1994). Context-Aware Computing Applications. *IEEE Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA, USA, 85-90.

Toivonen, S., Kolari, J., & Laakko, T., USA. (2003). Facilitating Mobile Users with Contextualized Content. *Artificial Intelligence in Mobile System Workshop*, Seattle, WA, USA, October,

Vetro, A., & Timmerer, C. (2005). Digital Item Adaptation: Overview of Standardization and Research Activities. *IEEE Transactions on Multimedia*. 7(3): 418-426.

W3C. "*Document Object Model (DOM)*." from http://www.w3.org/DOM/.

W3C. (2004). "*OWL-S: Semantic Markup for Web Services*." from http://www.w3.org/Submission/OWL-S/.

Wang, Y., Kim, J.-G., Chang, S.-F., & Kim, H.-M. (2007). Utility-Based Video Adaptation for Universal Multimedia Access (UMA) & Content-Based Utility Function Prediction for Real-Time Video Transcoding. *IEEE Transactions on Multimedia*. 9(2): 213-220.

Xie, X., Liu, H., Ma,.W -Y., & Zhang, H.-J. (2006). Browsing Large Pictures Under Limited Display Sizes. *IEEE Transactions on Multimedia*. 8(4): 707-715.

Yang, S.J.H., Tsai, J.J.P., & Chen, C.C. (2003). Fuzzy Rule Base Systems Verification Using High Level Petri Nets. *IEEE Transactions on Knowledge and Data Engineering*. 15(2): 457-473.

Yang, S.J.H., Zhang, J., Chen, R.C.S., & Shao, N.W.Y. (2007). A Unit of Information-Based Content Adaptation Method for Improving Web Content Accessibility in the Mobile Internet. *Electronics and Telecommunications Research Institute (ETRI) Journal*. 29(6):

794-807.

Yang, S.J.H., Zhang, J., & Chen, I.Y.L. (2008). A JESS enabled context elicitation system for providing context-aware Web services. *Expert Systems with Applications*, 34(4). 2254-2266.

Yang, S.J.H., Zhang, J., Tsai, J.J.P., Huang, A.F.M. (2009). SOA-based Content Delivery Model for Mobile Internet Navigation. *International Journal of Artificial Intelligence Tools*, 18(1), 141-161.

Zhang, J., Zhang, L.-J., Quek F., & Chung, J.-Y. (2005). A Service-Oriented Multimedia Componentization Model. *International Journal of Web Services Research (JWSR)*. 2(1): 54-76.

Zhou, C., Chia L.-T., & Lee, B.-S. (2005). Web Services Discovery on DAML-QoS Ontology. *International Journal of Web Service Research (JWSR)*. 2(2): 47-70.

Stephen J.H. Yang received his PhD degree in Computer Science from the University of Illinois at Chicago in 1995. He is now a Professor of the Department of Computer Science and Information Engineering and the Associate Dean of Academic Affairs, National Central University, Taiwan. He is the co-founder and the CEO of T5 Corp, a company providing XML-based Web services. Dr. Yang has published 2 books and over 150 journal articles and conference papers. He is currently the Co-Editor-in-Chief of the International Journal of Knowledge Management & E-Learning, and in the advisory board of the Educational Technology and Society; and in the editorial boards of the International Journal of Web Services Research and the International Journal of Knowledge and Learning. He severed as the Program Co-Chair of IEEE MSE 2003, IEEE CAUL 2006, and IEEE W2ME 2007. His research interests include Web services, Web 2.0, software engineering, knowledge engineering, semantic Web, and context aware ubiquitous computing. He is a member of IEEE.

Jia Zhang received her Ph.D. in Computer Science from University of Illinois at Chicago in 2000. She is now an Assistant Professor of Department of Computer Science at Northern Illinois University. Zhang has published 1 book titled "Service Computing" and over 80 technical papers in journals, book chapters, and conference proceedings. She is an Associate Editor of the IEEE Transactions on Services Computing (TSC), International Journal of Web Services Research (JWSR), and the Advances in Web Services Research (AWSR) Book Series, Idea Group. Zhang serves as Program Vice Chair of IEEE International Conference on Web Services (ICWS 2008, 2007 & 2006). Her current research interests center around Services Computing. She is a member of the IEEE.

Jeff J.S. Huang is a PhD student at the Department of Computer Science and Information Engineering, National Central University, Taiwan. His research interests include Web 2.0, CSCW, and CSCL.

Jeffrey J.P. Tsai received his PhD degree in Computer Science from the Northwestern University. He is now a Professor of the Department of Computer Science at the University of Illinois at Chicago. He is also a Senior Research Fellow at the University of Texas at Austin. Tsai coauthored 7 books and has published over 150 journal articles and conference papers. He serves as the General Co-Chair of the 16th IEEE International Symposium on Software Reliability Engineering, the 7th IEEE International Symposium on Multimedia, the 1st IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, the 3rd

International Conference on Ubiquitous Intelligence and Computing, and the 5th IEEE International Symposium on Bioinformatics and Bioengineering. Tsai is currently the Co-Editor-in-Chief of the International Journal on Artificial Intelligence Tools and an Associate Editor of the IEEE Transactions on Services Computing. He was an Associate Editor of IEEE Transactions on Knowledge and Data Engineering, and sits on many editorial boards of international journals. His current research interests include sensor networks, ubiquitous computing, trustworthy computing, intrusion detection, software reliability and adaptation, knowledge-based software engineering, distributed real-time systems, multimedia systems, and intelligent agents. He is a Fellow of the IEEE, the AAAS, and SDPS.