A Context-driven Content Adaptation Planner for Improving Mobile Internet Accessibility

Stephen J.H. Yang[1], Jia Zhang[2], Angus F.M. Huang[3], Jeffrey J.P. Tsai[4], Philip S. Yu[5]
[1, 3]*Dept. of Computer Science and Information Engineering, National Central University, Taiwan*
[2]*Department of Computer Science, Northern Illinois University, USA*
[4,5]*Department of Computer Science, the University of Illinois at Chicago, USA*
[1]*jhyang@csie.ncu.edu.tw,*[2]*jiazhang@cs.niu.edu,*[3]*fmhuang@csie.ncu.edu.tw,*
[4]*tsai@cs.uic.edu,* [5]*psyu@cs.uic.edu*

Abstract

This paper presents our design and development of a context-driven content adaptation planner, which 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 (DLs) to formally define context profiles and requirements and automate content adaptation decision. In addition, the computational overhead caused by content adaptation can be moderately decreased through the reduction of the size of adapted content.

Key words: Adaptation rules, Content adaptation, Description logics, Mobile Internet, Context.

1. 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, people tend to continue to work while on the move; their residing environments thus may change constantly and Web content delivery should also subject to the changes for better performance. For example, if a user moves into a blurred environment (e.g., due to 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 on 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, which has impact on content delivery and presentation including receivers' personal profiles, receiving devices, communication network, location, activity, and time [1][2][3][4][5].

Expected 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 some content adaptation techniques [6][7][8][9][10][11] between specific multimedia types, such as between images and video, some other researchers focus on exploring how to conduct proper content adaptation based on receiving contexts [12][13][14]. Although the literature has witnesses these effective content adaptation efforts and techniques, these works typically do not support automatic content adaptation decision; nor do they support configurable and extensible contextual environment specifications. We argue that a key to the issues is to build a formal foundation for the area.

In contrast with the previous works lacking a clear semantic basis, this research intends to study a semantic foundation for content adaptation. In our best 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. Our method could also reduce content access time. For example, if a user is accessing a film while driving, then the context-driven content

978-0-7695-3310-0/08 $25.00 © 2008 IEEE
DOI 10.1109/ICWS.2008.31

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 then presented in Section 2. We formalize context profile and requirement definition using description logics in Section 3. A context-driven content adaptation planner will be presented in Section 4. We conclude this paper in Section 5.

2. Related work

He et al. [14] identify three types of objects from an HTML page, namely, structure, content, and pointer objects. Mukherjee et al. [7] propose to associate content with metadata defining adaptation choices and their resulting media characteristics. Structural information of a Web page and the inter-relationships between presentation objects are managed by interactions and relationships between identified objects. 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 are higher.

Adaptation rules, or policies, are typically used to guide content adaptations. Kinno et al. [15] adopt policy descriptions to designate how an adaptation engine should behave according to changing environments. Lemlouma and Layaida [16] and Phan et al. [17], on the other hand, both design application-specific adaptation engines without employing adaptation policy descriptions. In contrast with their work based on stable and predefined content adaptation rules, our research applies DLs to formalize context and requirement specifications and automate content adaptation decision making.

A number of context-based adaptation methods [18][12][19][20][21][16][22] are proposed to customize Web content according to client contextual environments, including personal preferences, device capabilities, and access environments. Julien and Roman [23] propose a *view* concept to represent application-specific contextual information. Bellavista et al. [24] adopt metadata for representing context characteristics at a high level of abstraction. Cabri et al. [25] propose a two-dimensional model to describe the location information in mobile context: a physical location in space and a logical location within a distributed group or application. Nam et al. [8], ferences and perceptual characteristics as basis for effective visual content adaptation. Mukherjee et al. [4] take into consideration run-time conditions about terminal, network, user preference, and rights. Krause et al. [22] believe that contextual information include users' states and surroundings. A user's state can be extracted from the user's activities, 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.

As a fundamental technique in the field of semantic Web, Description logics (DLs) support formal knowledge representation in a structured manner [26]. W3C-endorsed OWL [27] is built on top of DLs. However, OWL is built 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.

3. Formalization of context specifications and decisions

3.1. 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 [26]. 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 [27].

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.

3.2. 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 for appropriate content adaptation and delivery.

3.3. Context requestor layer

Without losing generality, in this research, we consider four aspects of a receiving context environment: 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.

$ContextProfile \subseteq T$

$Identify \subseteq ContextProfile$

$Condition \subseteq ContextProfile$

$Network \subseteq ContextProfile$

$Device \subseteq ContextProfile$

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 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.

$What \subseteq Condition; Where \subseteq Condition;$

$What \subseteq Condition; Location \subseteq Condition;$

$CommunicationProtocol \subseteq Network;$

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.

3.4. 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 [28]; 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 specification is defined in ContextPrecondition. Furthermore, some side effect might be stated (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.

3.5. An example of using DLs to define context requirements and environments

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

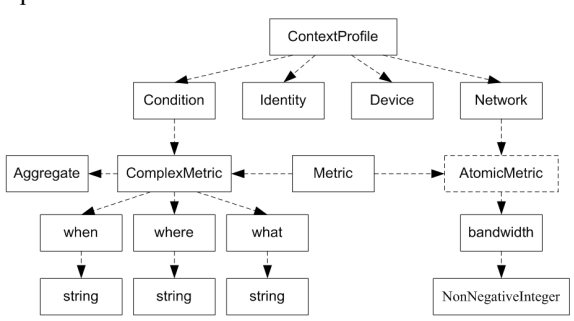

Figure 1. Part of context ontology.

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)

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

$$\cap(\text{"}meeting\text{"}\, what.WhatStringMetric)$$

$$\cap(\text{"}03/09/2008\text{"}\, when.WhenStringMetric)$$

$$\cap(\text{"}meetingroomA306\text{"}\, where.WhereStringMetric)$$

3.6. 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 is 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: $a \le x(i) \le b$, where $a \le b$, and a and b are both non negative integers.

Axiom 1. For two constraints, x and y, on the same context attribute *i*,

$a \le x(i) \le b, c \le y(i) \le d, a \le b, c \le d$,

$x(i) \succ y(i)$, iff $c \le a$ and $b \le d$.

For simplicity, we consider that one constraint on a context attribute only specify 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 ($\lhd$) 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

$P = x(C) \bigcap x(D) \bigcap x(N), R = y(C) \bigcap y(D) \bigcap y(N)$

, $P \lhd R$, 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 $\lhd$ has transitive property:

- $(P \lhd R) \wedge (R \lhd Q) \Rightarrow (P \lhd Q)$

Proof: These propositions are straightforward derived from their definitions.

4. Context-driven content adaptation planner

Based on our DL-based context formalization, we have designed a rule-based planner for supporting context-driven content adaptation. 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.

4.1. Web page decomposition and object management

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

4.2. 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. 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 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.

4.3. 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.

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.

```
;; 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))))
```

Figure 2. Content adaptation rules written in JESS

As mentioned in Web page decomposition phase, the inter-object relationships (pre-order or post-order sequences) will be used to restore an objects' 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.

4.4. Rule-based adaptation management using JESS

In our research, we adopt OWL to build our ontology, as OWL is based on DLs and is natural for us to define our contexts and requirements. We have adopted JESS (JESS) (Jess Expert System Shell), a rule engine for the Java platform, to design and implement content adaptation rules. Examples of rules considering condition are listed in Figure 2. 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).

4.5. Implementation results

We have implemented a content adaptation planner. Figure 3 illustrates some execution results of our content adaptation planner with a comparison of Web browsing between conventional and handheld devices. The middle of Figure 3 shows the yahoo home page (http://www.yahoo.com/) on a desktop browser. The left-hand side of Figure 3 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 3 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 3, 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.

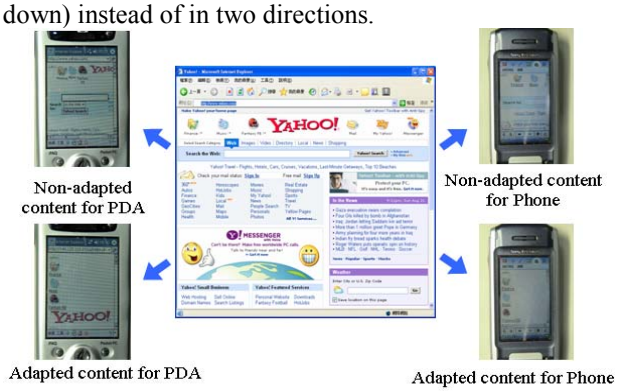

Figure 3. Snapshots of adapted Web page for NB, PDA, and Phone.

5. 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.

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 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. 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.

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

This work is supported by National Science Council, Taiwan under grants NSC 95-2520-S-008-006-MY3 and NSC96-2628-S-008-008-MY3.

References

[1] B. N. Schilit, N. I. Adams, and R. Want. "Context-Aware Computing Applications." IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, USA, 85-90, 1994.

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

[3] M. Satyanarayanan. "The Many Faces of Adaptation." IEEE Pervasive Computing, 3(3), Jul.-Sep., 4-5, 2004.

[4] D. Mukherjee, E. Delfosse, J.-G. Kim, and Y. Wang. "Optimal Adaptation Decision-Taking for Terminal and Network Quality-of-Service." IEEE Transactions on Multimedia, 7(3), Jun., 454-462, 2005.

[5] C. Julien and G.-C. Roman. "EgoSpaces: Facilitating Rapid Development of Context-Aware Mobile

Applications." IEEE Transactions on Software Engineering, 32(5), May, 281-298, 2006.

[6] R. Mohan, J. R. Smith, and C. S. Li. "Adapting Multimedia Internet Content for Universal Access." IEEE Transactions on Multimedia, 1(1), 104-114, 1999.

[7] D. Mukherjee, E. Delfosse, J.-G. Kim, and Y. Wang. "Optimal Adaptation Decision-Taking for Terminal and Network Quality-of-Service." IEEE Transactions on Multimedia, 7(3), Jun., 454-462, 2005.

[8] J. Nam, Y. M. Ro, Y. Huh, and M. Kim. "Visual Content Adaptation According to User Perception Characteristics." IEEE Transactions on Multimedia, 7(3), Jun., 435-445, 2005.

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

[10] X. Xie, H. Liu, W.-Y. Ma, and H.-J. Zhang. "Browsing Large Pictures Under Limited Display Sizes." IEEE Transactions on Multimedia, 8(4), Aug., 707-715, 2006.

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

[12] W. Y. Lum, and F. C. M. Lau. "A Context-Aware Decision Engine for Content Adaptation." IEEE Pervasive Computing, 1(3), 41-49, 2002.

[13] Z. Hua, X. Xie, H. Liu, H. Lu, and W.-Y. Ma. "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), Dec., 1650-1662, 2006.

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

[15] A. Kinno, H. Yukitomo, and T. Nakayama. "An Efficient Caching Mechanism for XML Content Adaptation." the 10th International Multimedia Modeling Conference (MMM'04), Jan., 308-315, 2004.

[16] Lemlouma, T. and Layaida, N. (). "Context-Aware Adaptation for Mobile Devices." 2004 IEEE International Conference on Mobile Data Management, 106-111, 2004.

[17] T. Phan, G. Zorpas, and R. Bagrodia. "An Extensible and Scalable Content Adaptation Pipeline Architecture to Support Heterogeneous Clients." the 22nd International Conference on Distributed Computing Systems, 507-516, 2002.

[18] Z. Lei, and N. D Georganas. "Context-based Media Adaptation in Pervasive Computing." Canadian Conference on Electrical and Computer Engineering, May, 2001.

[19] A. Pashtan, S. Kollipara, and M. Pearce. "Adapting Content for Wireless Web Service." IEEE Internet Computing, 7(5), 79-85, 2003.

[20] S. Toivonen, J. Kolari, and T. Laakko. "Facilitating Mobile Users with Contextualized Content." Artificial Intelligence in Mobile System Workshop, Seattle, WA, USA, October, 2003.

[21] B. Kurz, I. Popescu, and S. Gallacher. "FACADE - A Framework for Context-Aware Content Adaptation and Delivery." Second Annual Conference on Communication Networks and Services Research, 46-55, 2004.

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

[23] C. Julien, and G.-C. Roman. "EgoSpaces: Facilitating Rapid Development of Context-Aware Mobile Applications." IEEE Transactions on Software Engineering, 32(5), May, 281-298, 2006.

[24] P. Bellavista, A. Corradi, R. Montanari, and C. Stefanelli. "Context-Aware Middleware for Resource Management in the Wireless Internet." IEEE Transactions on Software Engineering, 29(12), Dec., 1086-1099, 2003.

[25] G. Cabri, L. Leonardi, M. Mamei, and F. Zambonelli. "Location-Dependent Services for Mobile Users." IEEE Transactions on Systems, Man and Cybernetics, Part A, 33(6), Nov., 667-681, 2003.

[26] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider. The Description Logic Handbook: Theory, Implementation, Applications. Cambridge, UK, Cambridge University Press, 2003.

[27] W3C. "OWL-S: Semantic Markup for Web Services." from http://www.w3.org/Submission/OWL-S/, 2004.

[28] C. Zhou, L.-T. Chia, and B.-S. Lee. "Web Services Discovery on DAML-QoS Ontology." International Journal of Web Service Research (JWSR), 2(2), Apr.-Jun., 47-70, 2005.

[29] S. J. H. Yang, J. J. P. Tsai, and C. C. Chen. "Fuzzy Rule Base Systems Verification Using High Level Petri Nets." IEEE Transactions on Knowledge and Data Engineering, 15(2), Mar.-Apr., 457-473, 2003.

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