An XML-based Multimedia Middleware for Mobile Online Auctions

Matthias Wagner, Wolf-Tilo Balke, Werner Kießling

Angaben zur Veröffentlichung / Publication details:

Wagner, Matthias, Wolf-Tilo Balke, and Werner Kießling. 2001. "An XML-based Multimedia Middleware for Mobile Online Auctions." Augsburg: Universität Augsburg.

UNIVERSITÄT AUGSBURG

An XML-based Multimedia Middleware for Mobile Online Auctions

Matthias Wagner, Wolf-Tilo Balke, Werner Kießling

Report 2001-2 Januar 2001

INSTITUT FÜR INFORMATIK

D-86135 AUGSBURG

Institut für Informatik
Universität Augsburg
D–86135 Augsburg, Germany
http://www.Informatik.Uni-Augsburg.DE

An XML-based Multimedia Middleware for Mobile Online Auctions

Matthias Wagner, Wolf-Tilo Balke, Werner Kießling

Chair of Database and Information Systems
Institute of Computer Science, University of Augsburg
{*wagner,balke,kiessling*}*@informatik.uni-augsburg.de*

Abstract

Pervasive Internet services today promise to provide users with a quick and convenient access to a variety of commercial applications. However, due to unsuitable architectures and poor performance user acceptance is still low. To be a major success mobile services have to provide device-adapted content and advanced value-added Web services. Innovative enabling technologies like XML and wireless communication may for the first time provide a facility to interact with online applications anytime anywhere. We present a prototype implementing an efficient multimedia middleware approach towards ubiquitous value-added services using an auction house as a sample application. Advanced multi-feature retrieval technologies are combined with enhanced content delivery to show the impact of modern enterprise information systems on today's e-commerce applications.

Keywords: mobile commerce, online auctions, middleware architectures, pervasive Internet technology, multimedia database applications.

1 Introduction

Emerging standards and technologies for mobile Web services promise to create Internet ubiquity. Allowing users not only to carry their data and user profiles with them, but also guaranteeing facilities to interact with online applications anytime anywhere they are expected to overcome today's Internet problems by providing users with a quick and convenient way to shop, pay bills, or find places and products that they want or need. However, the industry is still struggling how to best take advantage of the Internet in the wireless environment: while simple mobile applications like the Short Messaging System (SMS) or e-mail over cellular phones are considered as a success, the recent roll-out of mobile services based on the Wireless Application Protocol (WAP) by European telecommunication carriers has shown that new mobile commerce applications still lack consumer acceptance. Poor design and information architecture are hampering the usability of WAP sites. The lack of specially adapted multimedia content additionally decreases their acceptance by consumers.

To be a major success mobile services must target users with immediate, context-dependent content and advanced value-added Web services. General services like e-shopping or Internet search engines are less likely to succeed in the mobile environment unless major improvements happen. Mobile devices, be it WAP cellular phones or Internet enabled handheld PCs, might indeed support a broad success of e-commerce applications throughout the public. But, whereas existing WAP services are mainly designed as stand-alone applications we feel that the successful future of m-commerce lies in a truly pervasive approach. This means that existing Web services should be successively expanded to support access for the increasing number of mobile clients. In this paper we will show how to smooth the way from standard Internet services towards value-added m-commerce applications using the example of an online auction house as a value-added Web service enabled for ubiquitous access.

The developments in enterprise information systems and architectures [14] can also be useful in today's e-commerce applications. The main idea of an enterprise information system [1] is to provide a single, Web-based interface to sometimes incompatible data scattered across a variety of repositories. These systems thus provide one (possibly personalized) logical view and common query capabilities across different platforms and information sources. The information sources in general are not limited to relational databases, image archives or document repositories, but more sophisticated systems also allow external Web data to be directly integrated into query result sets [15]. In general the ideal goal is 'just in time' information, retrieved and assembled as needed, freely accessible and exchangeable across diverse systems. Though most enterprise systems provide suitable connectors for multi-repository support, especially when it comes to advanced federated searches and the delivery of user tailored information to mobile devices, even the most modern systems experience major problems. Approaches to advanced federated multi-feature searches [9, 10] and content delivery [17, 19] were recently presented in the literature. In this paper we show how these techniques can enrich an XML-based multimedia middleware in our prototypical application.

The rest of this paper is organized as follows. In section 2 we will deal with the demands of online auctions and present an example for user interaction in modern Web-based auction applications. User profiles and usage patterns are recognized as the basis for pervasive applications. By comparison to existing online auctions like Ebay or Yahoo!Auctions the presented system allows auctioneers to search auction catalogs by content-based and fuzzy full-text retrieval methods in addition to standard search criteria. The way our sample auction application is enabled for universal access through an XML-based middleware approach is explained in section 3. Section 4 deals with the nature of federated and multi-feature retrieval capabilities and its benefit for our application. Here we present the application of the Stream-Combine algorithm to leverage efficient and multi-feature search capabilities in multimedia resources. The delivery of result sets and more general content to a wide variety of possibly mobile clients is finally discussed in section 5. We conclude with a short summary and an outlook on future work.

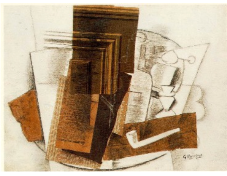
Rank	Image	Description	Current Bid
1		**Category: Painting** This beautiful modern oil on canvas painting by Pablo Picasso is entitled ’Woman in an Armchair’ and was created in the year 1927. It presents a woman, who occupies an armchair upholstered in a leaf-printed fabric...	**$ 3,560**
2		**Category: Painting** ’Bottle, Newspaper, Pipe, and Glass’ is the title of this painting by Modernist painter Georges Braque. Its unique style is supported by his use of charcoal and various papers pasted on paper...	**$ 1,230**
⋮	⋮	⋮	⋮

Figure 1: A sample result set for a complex query.

2 Mobile online auctions

Today’s online auction houses like Ebay or Yahoo!Auctions offer a wide variety of categories to choose from. Items are presented together with a short description and a digitized photo. However, the search capabilities of these auction applications only focus on standard attributes like item numbers, categories, keywords and price ranges. More sophisticated search capabilities like weighted searches or visual retrieval are currently not supported.

We claim that the acceptance of e-commerce applications will increase with the development of effective multimedia search facilities allowing users to compose complex queries in an intuitive way. The better customer queries can be composed the more satisfying the retrieval results will generally be. Moreover, anticipating typical usage patterns and user profiles will further help to get a broad acceptance.

2.1 Query profiles

Let us consider a short example of a next-generation online auction house. Assume that a customer named Frank is interested in buying a modern painting to furnish a new apartment. Using our online auction he will state

- the category ’painting’,

- the description of the painting should indicate that the painting is 'modern' art
- and to go well with the wallpaper, carpets and furniture brown or earthen colors should be prevalent.

In this example the auction query engine will have three aspects to satisfy. For each aspect a ranked result has to be computed using (possibly different) suitable data sources. At the end a few overall best objects matching the different aspects best are calculated and returned to the user as shown in figure 1.

The typical user interaction with an auction system will first provide some query information for the search together with a user profile for the later presentation of the results. The auction system will analyze the query constraints and (depending on the user's profile) decide which sources to query. Having merged all ranked result sets the overall top-scored objects can be returned. Consider for our example two different profiles. Dealing with pieces of art our auction house will have several typical groups of customers. To understand how our architecture deals with their typical profiles let us choose two users Frank and Cathy. Frank is our above user who wants to furnish his apartment and looks for a modern painting which is mainly painted in brown colors. Cathy is a collector who is strongly interested in copies of paintings by Picasso that are oil on canvas technique. To fit in her collection she also would prefer a mostly brown painting, but it does not really matter to her. We will be concerned with these queries more deeply in section 4. For now let us assemble the two different query profiles in in the table of figure 2.

query profiles	**Frank**	**Cathy**
category	painting	painting
painter	don't care	Picasso
technique	don't care	oil on canvas
style	modern	don't care
average color	brown	brown

Figure 2: Query profiles for different users.

2.2 Usage profiles and mobile access

As wireless technology improves at an ever-increasing rate, mobile devices for every situation and user are being created. Let us focus on two main types of such devices:

Personal Digital Assistants: The personal digital assistant (PDA), e.g. the Palmpilot, has become the standard handheld computer. With Internet-enabled PDAs users can have real-time connections to text information and low-resoluted images.

WAP Phones: Wireless Application Protocol (WAP) [20] is designed to create ubiquitous access to Internet resources. Using WAP-enabled mobile phones, users can mainly perform simple text-based tasks.

Of course, for the delivery of the result sets the technical environment of the user has to be taken into account. On one hand users may at home or in the office be connected by a LAN using a high speed PC, on the other hand the user may only use a WAP mobile phone having a rather slow connection being out of home, e.g. riding the train. Users having high-speed access would obviously process a large compound document containing e.g. all the images, whereas users with slow access probably would rather prefer a text-only version of the document or even be satisfied by an abstract of the document. These compound documents not only have to be adequate for the device, but also are more or less useful in different stages of the auction.

Since different tasks may need different modes of access, the table in figure 3 shows typical usage patterns together with the devices needed to perform a task. Tasks may differ in their frequency of performance, the location of the user and the need for more or less advanced technical devices. The most important thing is to enable users to perform frequently needed tasks with preferably small and mobile devices. Thus, these tasks can be integrated into their daily routine with a maximum degree of freedom. Knowing the query profiles, usage patterns and technical restrictions a user profile can be modeled [4].

usage patterns	user device			frequency	location
browse, choose item	PC	-	-	rare	home
find similar item	PC	PDA	-	rare	anywhere
check, increase bid	PC	PDA	WAP	often	anywhere

Figure 3: Usage patterns for different devices.

Again consider our above example. To choose a painting from the auction catalog one would definitely prefer to get high resoluted color images and descriptive text. Thus a typical task is browsing catalogs via a Web browser using a PC having fast Internet access. Once having decided to bid for any object it is useful to follow the auction and if necessary to increase the bid. Since online auctions may take a few days time, it is necessary to check the current bid from time to time. However, for this task not all the descriptive and visual information is needed, but only e.g. the catalog number, title or painter. Simple, but frequent tasks like this can thus conveniently be carried out anywhere using a mobile phone or PDA, if only few data has to be exchanged. But advanced mobile devices like PDAs even have the capabilities of performing slightly more complex tasks. For instance if the bids for a chosen object are above a certain limit, users might want to search for objects most similar to the previously chosen one. Suppose for the course of our example that both sample users have access to the Internet via a standard Web browser. In addition Cathy owns a WAP phone, whereas Frank uses a PDA for mobile access.

3 The XML-based multimedia middleware

Due to the rapid pace of technological change, emerging standards and the constant evolution of mobile devices, the use of flexible and adaptable technology and methodologies is essential for pervasive applications. We think that content management [11] is paramount for pervasive applications, meaning that application logic, server-side and database-side content as well as delivery formats are to be strictly separated. This section illustrates how we support the virtues of content management.

3.1 XML base technologies

To address the requirements of commercial Web applications and enable the further expansion of Web technology into new domains, the World Wide Web Consortium has developed the Extensible Markup Language (XML) [5]. Aiming at a middleware approach our prototypical implementation builds upon the following XML standards, tools and extensions.

XML (Extensible Markup Language): XML is a subset of SGML (Standard Generalized Markup Language) and basically a markup language for documents containing structured information. XML is currently on the way to become the standard markup language for Internet applications.

XSL (Extensible Stylesheet Language): XSL and XSLT (XSL Transformations) [12] provides a powerful implementation of a tree-oriented transformation language for converting instances of a specific XML document type into basically any document format, like either simple text, the legacy HTML vocabulary, or document formats currently used in mobile applications, e.g. WAP.

XSP (Extensible Server Pages): XSP [8] is a new technology integrated into the Apache Web server for building Web applications based on dynamic XML content. They allow XML developers and designers to rapidly develop and easily maintain dynamic Web pages that leverage XML applications. Related technologies for server-centered computing of dynamic content include Sun's JavaServer Pages or Oracle's XSQL servlet.

Java Technology: The XSP technology connects underlying databases and application modules using the Java programming language. The application logic resides in middleware resources, e.g. database access, database search and the combination of search results is coded in Java and executed in a middle-tier java virtual machine. Furthermore Apache Web server with its servlet engine JServ is used to handle incoming client search requests and to initiated content delivery.

3.2 An advanced multimedia middleware architecture

Figure 4 depicts the conceptual design of our system. How the XML technologies mentioned above come into play is best explored by a short example. Let us consider a sample auction session:

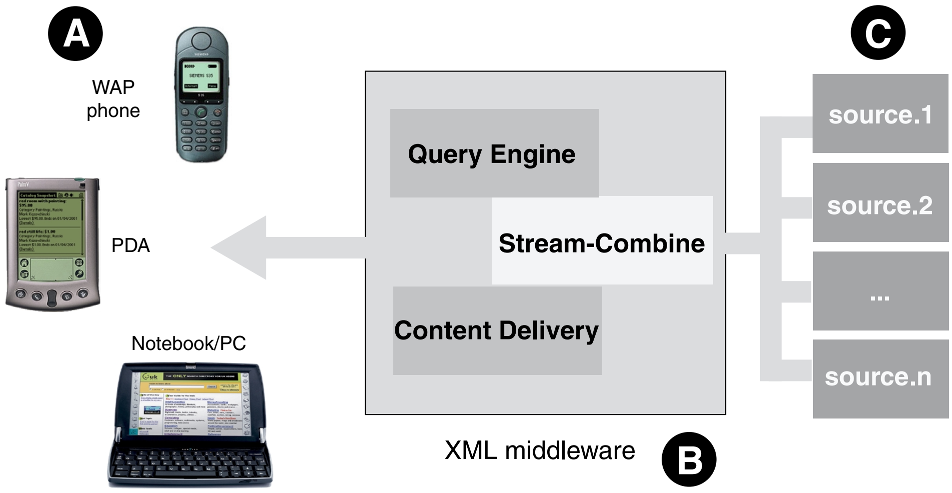


Figure 4: An XML-based multimedia middleware architecture.

1. Cathy carrying her WAP phone (A) wishes to track several items from the auction catalog. The first step in initiating a database search from any device is to download a search form from the application server. A corresponding client-side request from (A) to the middle-tier application server (B) results in preparing a generic XML form, provided by the Query Manager, for delivery to the client.

2. Therefore the server-side search form is passed to the Delivery Engine together with a client profile. XSL rendering is then done on the XML form w.r.t. the given profile and results in a client specific search form which is in turn delivered to the device (A).

3. A search request is issued from the client (A) by completing the search form and sending it back to the Query Manager (B).

4. Complex queries are allowed to be constructed via search forms and are typically restricted to the best k results fulfilling the request. As will be explained in section 4 these complex queries are split into their atomic parts and passed to the underlying data sources (C) for execution. The combining algorithm Stream-Combine (B) takes over and is responsible for the efficient combination of the possibly ranked result streams from the sources.

5. Within the middleware (B) the search result is then encapsulated in a generic XML document and again passed to the Delivery Engine. As with the search form the XML result set is finally transcoded to a format that will satisfy the client and delivered to Cathy's device (A).

The parts of the architecture presented here will be discussed in depth in the following sections. The next sections deals with combing algorithms and federated multi-feature searches, whereas section 5 presents the techniques used for multimedia delivery.

4 Federated searches and multi-feature queries

The ability to perform federated searches and multi-feature queries is one of the most demanding in enterprise information systems. The answer sets will generally consist of compound documents involving data of different types collected from different sources. These searches include:

- Traditional full-text searches.
- Fuzzy search capabilities including phonetic search or thesauri.
- Attribute-based queries relying on syntactical attributes, e.g. documents of certain length or images in specific image formats.
- Advanced content-based retrieval relying on semantical attributes, like semantic document retrieval or query by visual example.

Since techniques for multi-feature queries have already shown their usefulness in state of the art information systems [2], we will adapt these techniques for the use in our auction system. Our above example will again help us to understand the way federated searches and multi-feature queries can be performed.

For user Frank we have to pose a complex query involving three aspects or features: The category of object is 'painting', the object should be in 'modern' style and the color should mainly be brown. All these query models are essentially different. Whereas the category can be evaluated as an exact match query, the query on the keyword 'modern' will return a ranked result set of descriptions. The query on color even has to be performed using visual retrieval returning a ranked set of documents containing more or less brown images. Using the emerging SQL/MM standard [6] this query can be expressed as follows:

```
1  SELECT top(5, images)
2  FROM    repository
3  WHERE category = 'painting'
4  RANK BY description CONTAINS 'modern'
5        AND SI_average_color(images) SIMILAR TO (150,120,20)
```

In this sample query statement

- the top five images are retrieved (line 1)

- that belong to the category 'painting' (line 3),
- have a description which contains the word 'modern' (line 4) and
- have an average color close to the RGB-value (150,120,20) which is a shade of brown (line 5).

Since in the expression above no weights are specified, all these parts of the query are considered equally weighted. As commercial database systems currently do not yet support the full SQL/MM standard, in our auction system prototype this statement is further mapped onto the specific implementations of the underlying database management systems. The query result in this case consists of three parts, an exact match part for the category, a ranked result list for the keyword 'modern' in the description, and a ranked result list for the visual retrieval part. Efficiently combining these different result sets to get the overall best objects has been a major performance bottleneck until recently. With Quick-Combine [9] and Stream-Combine [10] two efficient combining algorithms have been recently developed that can improve efficiency by 1–2 orders of magnitude.

In our prototypical auction system the Stream-Combine algorithm is used for merging multi-feature result lists within the middleware even from distributed data sources. Stream-Combine guarantees the correct retrieval of the k overall top-ranked results. For score aggregation any monotonic combining function can be used, including weighted queries. Due to its heuristic control flow this algorithm is narrowly self-adapting to the particular score distributions. Top-ranked results can be computed and output incrementally. The way how results are merged obviously is strongly user dependent. In complex queries the user might want to emphasize some properties, i.e. weighting the query, or use a special combining function. Such functions can be for instance arithmetic means to ensure a certain consistency of performance for the top-scored objects in all query parts or maximum/minimum to find the top performers in any query aspect. Thus not only the query aspects themselves, but also user preferences as emphasis on certain topics or preferred combining functions add semantic information for a relevant search result. Though using the same auction system users having different profiles should generally be allowed to use different combining functions to express their search semantics.

Let us have a short glance at the sample combining functions of Cathy and Frank. The score for any auction item X consisting of an image together with a category and a description can for instance be computed as:

$$\begin{aligned} score_{\text{Cathy}}(X) &= \frac{1}{5}\,((2 \cdot score_{\text{category='painting'}}(X) + \\ &\quad 2 \cdot score_{\text{description CONTAINS ('Picasso' AND 'oil on canvas')}}(X) + \\ &\quad score_{\text{SI\_average\_color(image) SIMILAR TO (150,120,20)}}(X)) \\ score_{\text{Frank}}(X) &= \frac{1}{3}\,(score_{\text{category='painting'}}(X) + \\ &\quad score_{\text{description CONTAINS 'modern'}}(X) + \\ &\quad score_{\text{SI\_average\_color(image) SIMILAR TO (150,120,20)}}(X)) \end{aligned}$$

We know that Cathy is a collector of paintings by Picasso. Obviously she will assign very high weights to the category and keyword search, and only rather low values to her preferred brown color. She will even be pleased with the evaluation of the category as a hard filter condition like it is provided by today's database systems. Frank on the other hand wants to furnish his apartment. Thus the color and appearance of the painting is of high importance to him. He will probably assign equal weights to all categories. But in his case it might also be sensible to weaken the filter effect of the category. Since he wants to cover a wall, for instance he could also be satisfied by a modern, brown tapestry. Unlike traditional databases our multimedia middleware can easily adapt to these requirements using the Stream-Combine approach.

Related database applications and middleware approaches like GARLIC [3] or HERON [13] have already started to use the capabilities of such an advanced retrieval. However, if ranked result sets from distributed data sources already have to be merged within a retrieval system, the combination does not have to take place inside the middleware. Consider for example Cathy's query. She wants the painting to be an oil on canvas technique by Picasso. Since both keywords have to be contained in the object's description, those two result sets can already be combined within the text-retrieval engine and passed on to the middleware as a single ranked result set. For those special combining tasks within a retrieval engine the Quick-Combine algorithm [9] is a prime candidate. Quick-Combine shows the same useful characteristics as Stream-Combine, but is especially efficient inside a retrieval engine or database kernel.

5 Multi-platform content delivery

Content management for pervasive applications needs to address both the nature of the content and the capabilities of the various client platforms the content is being delivered to. User preferences and technical limitations of the user devices are expressed by user profiles and usage patterns demanding different delivery formats.

Historically, electronic information was expressed, created, stored and transmitted by application-specific formats. Alternatively, with XML all data is structured without taking the later presentational layout into account. How our system benefits from the use of XML and XSLT is again best illustrated by our running example. As we have seen in section 4 both Frank and Cathy want to pose somehow similar queries against the auction database. Suppose that one evening, for initially searching the catalog, both use their standard Web browsers at home. The left hand side of figure 5 depicts the search form with enhanced multimedia input facilities for content-based search criteria completed for Frank's initial query. Let us assume furthermore that Frank and Cathy get the same top-scored item from the auction catalog and place a bid. During the next day Cathy follows the auction and occasionally checks the selected item using her mobile phone and increases her bid (cf. right hand side of figure 5).

In the middle-tier of our application, generic XML formats are used together with client-specific XSL stylesheets to tailor compound documents for any supported delivery platform. As we know from

Figure 5: Search request from a Web browser and tracking of bids from a WAP phone.

section 2 tracking of auction items is enabled for all existing usage patterns. Hence, when Cathy decides to check her selected item from her WAP phone she requests a download of the generic XML document displayed on the left hand side of figure 6. Besides a page title this generic XML document defines a search form having two input fields in universal style: the first one named 'category' is defined as a selection on the existing categories whereas the second one named 'usr' defines a text input field to submit the name of an auctioneer. This XML document is basically the same for any enabled platform. But since Cathy's request is issued from a WAP phone the application server will transform this generic search form from generic XML into WML prior to delivery (cf. right hand of figure 6).

```
<page>
  <title>Track Item</title>
  <form action="tracking.xml">
    <select name="category" dname="Category">
      <option value="">All</option>
      <option value="Carpet">Carpet</option>
      <option value="Paintings">Paintings</option>
      <option value="Porcelain">Porcelain</option>
    </select>
    <input name="usr" format="*a" dname="User" value=""/>
  </form>
</page>
```

```
<wml>
  <card id="top" title="Track␣Item">
    <p align="left">
      User: <input format="*a" name="name" value=""/>
      Category: <select name="category">
        <option value="">All</option>
        <option value="Carpet">Carpet</option>
        <option value="Paintings">Paintings</option>
        <option value="Porcelain">Porcelain</option>
      </select></p>
    <do label="start" type="accept">
      <go href="tracking.xml" method="get">
        <postfield name="category" value="$(category:noesc)"/>
        <postfield name="usr" value="$(name:noesc)"/>
      </go>
    </do>
  </card>
</wml>
```

Figure 6: A generic XML form and the corresponding WML search form.

Using different client-specific XSL stylesheets the server-side delivery scheme for XML documents is virtually the same for any content: based on the user profiles associated with client requests the middle-tier server will associate specific stylesheets with server-side XML documents and use their formating instructions to output new delivery documents. According to the different usage patterns from figure 3, stylesheets for the creation of WML (WAP phones), restricted HTML (PDAs) and full HTML/CSS (Web browsers) reside in the middle-tier. Considering Cathy's tracking request, the rule-driven transformation of XML content is partially illustrated in figure 7.

```
<xsl:template match="form">
  <p align="left">
    <xsl:for-each select="select">
      <xsl:value-of select="@dname"/>:
      <select>
        <xsl:attribute name="name"><xsl:value-of select="@name"/></xsl:attribute>
        <xsl:for-each select="option">
          <option>
            <xsl:attribute name="value"><xsl:value-of select="@value"/></xsl:attribute>
            <xsl:value-of select="."/>
          </option>
        </xsl:for-each>
      </select>
    </xsl:for-each>
    ⋮
  <do type="accept" label="start">
    <go method="get">
      <xsl:attribute name="href"><xsl:value-of select="@action"/></xsl:attribute>
      <xsl:for-each select="input|select">
        <postfield>
          <xsl:attribute name="name"><xsl:value-of select="@name"/></xsl:attribute>
          <xsl:attribute name="value">$(<xsl:value-of select="@name"/>)</xsl:attribute>
        </postfield>
      </xsl:for-each>
    </go>
  </do>
</xsl:template>
```

Figure 7: Transformation of a generic XML form to a WML search form.

Each transformation rule is expressed in the stylesheet as a template rule with a 'match' attribute. Here a rule matching the XML structure 'form' is defined. Once the 'form' tag is found in the XML source the rule fires and results in the execution of the formatting instructions in the rule's body. In this example, XSL instructions necessary to process several selection fields in a generic XML form are displayed. Note that in contrast to HTML the concept of forms is not defined in the current WML standard. Figures 6 and 7 demonstrate a useful way of incorporating this concept into WML and other non-HTML delivery formats.

Let's continue our example. Cathy's bid eventually gets accepted. By noon she owns a new Picasso for her collection. Frank who stopped placing bids during the auction, now has to look for a new piece of art to furnish his apartment. Using his Palmpilot PDA he queries the auction catalog for similar items. The auction system uses the properties of his primarily selected item to pose a new query and delivers the results to Frank's PDA. Again, prior to delivery the generic XML representation of the search result is transformed. This time a stylesheet for the creation of restricted HTML content satisfying the limited resources of Frank's PDA is used. The most similar item in the result set is a self-portrait of Pablo Picasso. The right hand side of figure 8 shows the display of his PDA. He immediately places a bid for this item and follows the auction at home in the evening again using his Web browser (cf. figure 8). After a few hours he also is the proud owner of a nice Picasso painting.

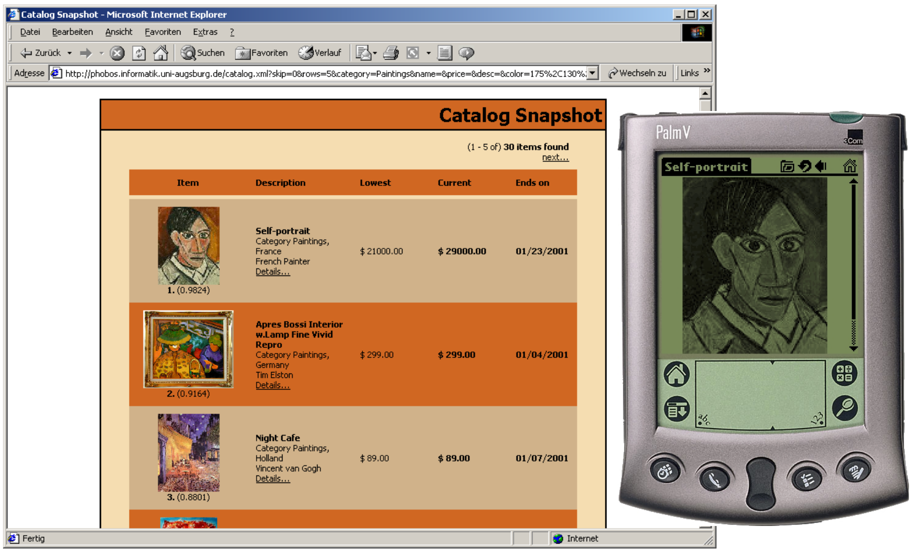


Figure 8: Search result in a Web browser and on the Palmpilot.

6 Summary and outlook

In this paper we presented a prototypical virtual auction house with ubiquitous access benefiting from modern information system technology. The implementation and evaluation was done using IBM DB2 Universal Database as the database management system for the underlying data sources. In particular DB2 Relational Extender technology [7] was used to support fuzzy full-text retrieval and content-based catalog queries. Additionally, due to the open design of the middle-tier application server, other database management systems and general data mediators supporting a ranked query model are easily adaptable.

Complex Internet applications will have to collect data from a variety of sources possibly using several retrieval engines. As we have illustrated in the discourse of this work the retrieval and aggregation strongly depends on user profiles and usage patterns. New efficient techniques like Stream-Combine [10] will enable efficient federated multi-feature searches w.r.t. specific query profiles. We proposed to build efficient middleware applications for mobile commerce using XML technology together with sophisticated algorithms for merging ranked result lists and pervasive content delivery. The running prototype will also be presented at the forthcoming international computer fair CeBIT 2001 in Hannover.

Besides, aggregated content is not uniquely accessed through standard Web browsers using high-performance network infrastructure but at a progressive rate through mobile devices. Existing Web applications lack the value added by enabling their services for multiple pervasive devices. Thus, content management and delivery in tune with federated and multi-feature searches promise to leverage future m-commerce applications. An advanced multimedia middleware architecture as presented in this paper efficiently supports these techniques in common Internet environments.

For multimedia delivery the text-based formats like XML, variations of HTML and WML, binary formats for the storage of multimedia data from auction catalogs can be taken into account. Delivery formats are not independent of each other but interrelated by conversion options. Given the diversity of today's image, audio or video formats this is even more true for multimedia content resulting in a large variety of ways to store and deliver multimedia data in various formats. The benefit of techniques for partitioning delivery data into formats that are physically stored in the middle-tier for fast delivery and those that are converted into a specific delivery format on-demand was shown in [18, 17]. Future work should also focus on XML-based standardized query languages, capable of accessing relational and object-oriented databases, together with multimedia extensions like SQL/MM or query languages for e-business. Recent XML applications like Motorola's VoxML [16] even offer standard capabilities for incorporating speech recognition and synthesis. Enriching the multimedia middleware presented in this paper with such advanced technologies will further improve the acceptance of m-commerce applications.

References

[1] H. Afsarmanesh, C. Garita, and L. Hertzberger. Virtual Enterprises and Federated Information Sharing. In *Proc. of the 9th Int. Conf. Database and Expert Systems Applications (DEXA 1998)*, LNCS Vol. 1460, pages 374–383, Vienna, Austria, 1998. Springer.

[2] W-T. Balke, U. Güntzer, and W. Kießling. Applications of Quick-Combine for Ranked Query Models. In *Proc. of the 1st DELOS Workshop on Information Seeking, Searching and Querying in Digital Libraries*, Zurich, Switzerland, 2000. DELOS.

[3] M. Carey, L. Haas, P. Schwarz, M. Arya, W. Cody, R. Fagin, M. Flickner, A. Luniewski, W. Niblack, D. Petkovic, J. Thomas, J. Williams, and E. Wimmers. Towards Heterogeneous Multimedia Information Systems: The GARLIC Approach. In *5th Int. Workshop on Research Issues in Data Engineering: Distributed Object Management*, pages 124–131. IEEE-CS, 1995.

[4] U. Cetintemel, M. J. Franklin, and C. L. Giles. Self-Adaptive User Profiles for Large-Scale Data Delivery. In *Proc. of the 16th Int. Conf. on Data Engineering*, San Diego, CL, USA, 2000. IEEE Computer Society.

[5] World Wide Web Consortium. Extensible Markup Language (XML) Version 1.0. W3C Recommendation, February 1998.

[6] P. Cotton, editor. *ISO/IEC FCD 13249-5:1999 SQL/MM SAF-005: Information Technology - Database Languages - SQL Multimedia and Application Packages - Part 5: Still Image*. ISO/IEC, 1999.

[7] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz. Efficient

and Effective Querying by Image Content. *Journal of Intelligent Information Systems*, 3:231–262, 1994.

[8] Apache Software Foundation. The Apache XML Project. http://xml.apache.org/cocoon, 2001.

[9] U. Güntzer, W-T. Balke, and W. Kießling. Optimizing Multi-Feature Queries for Image Databases. In *Proc. of the 26th Int. Conf. on Very Large Databases (VLDB 2000)*, pages 419–428, Cairo, Egypt, 2000.

[10] U. Güntzer, W-T. Balke, and W. Kießling. Towards Efficient Multi-Feature Queries in Heterogeneous Environments. In *Proc. of the IEEE Int. Conf. on Information Technology: Coding and Computing (ITCC 2001)*, Las Vegas, USA, April 2001. IEEE.

[11] R. Jain. Content-based multimedia information management. In *Proc. of the 14th Int. Conf. on Data Engineering*, pages 252–253, Orlando, FL, USA, 1998. IEEE Computer Society.

[12] Michael Kay. *XSLT Programmer's Reference*. Wrox Press, 2000.

[13] W. Kießling, K. Erber-Urch, W.-T. Balke, T. Birke, and M. Wagner. The HERON Project — Multimedia Database Support for History and Human Sciences. In *28. Annual Conf. of the GI: INFORMATIK98*, LNCS, pages 309–318, Magdeburg, Germany, 1998. Springer.

[14] W. Melling. Enterprise Information Architectures - They're Finally Changing. In *Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, pages 493–504, Minneapolis, Minnesota, USA, 1994. ACM.

[15] J. Widom R. Goldman. WSQ/DSQ: A Practical Approach for Combined Querying of Databases and the Web. In *Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, pages 285–296, Dallas, Texas, US, 2000. ACM.

[16] Motorola Wireless Resources. VoxML. http://www.motorola.com, 2001.

[17] J. R. Smith, R. Mohan, and C.-S. Li. Scalable Multimedia Delivery for Pervasive Computing. In *Proc. of the ACM Multimedia '99*, pages 131–140, Orlando, FL, USA, 1999. ACM Press.

[18] M. Wagner, S. Holland, and W. Kießling. Towards Self-tuning Multimedia Delivery for Advanced Internet Services. In *Proc. of the 1st Int. Workshop on multimedia intelligent storage and retrieval management (MISRM'99)*, Orlando, Florida, 1999.

[19] M. Wagner, S. Holland, and W. Kießling. Efficient and Flexible Web Access to Art-Historical Image Collections. In *Proc. of the 15th ACM Symposium on Applied Computing*, volume 2, pages 915–921, Como, Italy, 2000.

[20] Wireless Application Protocol Forum. *Official Wireless Application Protocol: The Complete Standard*. John Wiley & Sons, April 1999.