

The *x*POI Concept

Jens Krösche¹ and Susanne Boll²

¹ Oldenburg Research and Development Institute for Computer Science Tools and Systems (OFFIS), Escherweg 2, 26121 Oldenburg, Germany, +49 (0) 441 97 22 - 146, jens.kroesche@offis.de

² Carl von Ossietzky University Oldenburg, Faculty II, Department of Computing Science, Escherweg 2, 26121 Oldenburg, Germany, +49 (0) 441 97 22 - 213, boll@informatik.uni-oldenburg.de

Abstract. Today, most mobile applications use geo-referenced points of interest (POIs) on location-based maps to call the user's attention to interesting spots in the surroundings. The presentation of both, maps and POIs, is commonly location-based but not yet adapted to the individual user's needs and situation. To foster the user's information perception by emphasising the location-based information that is most relevant to the individual user, we propose the *x*POI concept – the modeling, processing, and visualisation of context-aware POIs. We introduce a data model for *x*POIs, supporting the exchange of *x*POIs and define an architecture to process and present *x*POIs in cooperation with a mobile information system. With the integration of context-awareness into POIs, we contribute to the development of innovative location- and context-aware mobile applications.

1 Introduction

Developments in areas like mobile computing, localisation, and wireless networks have recently increased the emergence of a wide range of mobile applications. One observation that can be made here, is that distinct tasks, such as navigation and orientation support through maps are part of nearly every mobile application. Typically, these applications provide not only a map but also a visualisation of geo-referenced information like points of interest (POIs). Although many applications provide similar features with regard to map and POI presentation, nearly all of them build their own proprietary mechanism to realise them. Furthermore, as the situation of a mobile user constantly changes, so should the visualisation of the POIs; behind every corner new impressions and situations – the context – could come up that might influence the user and should therefore be

reflected by the applications. Today, factors like role, location, and time play an increasingly important role when selecting and displaying information. However today’s applications provide only static POIs that do not adapt to the user’s context.

In this paper, we present a concept for context-aware points of interest (*x*POIs) that make POIs aware to the users context. The *x*POI concept allows to adapt the presentation of POIs according to a current user situation. This helps the user to focus on those POIs that are most “important” in a given situation, guiding the attention of the user to the needed information. Another benefit is the emergence of new interaction methods: Since *x*POIs are sensitive to the current context situation, they can adapt their visualisation, and in addition trigger interaction. Hence, the user is not restricted to a direct interaction with the system; rather an additional indirect interaction by changing his context situation is possible.

In the following we propose a data model for *x*POIs that reflects context-aware visualisation and interaction aspects. In addition to the *x*POI data model, we present a processing engine (*x*POI engine) which is able to analyse and process instances of this *x*POI data model and generate the appropriate *x*POI presentations. At the end, we illustrate the usage of the *x*POI-concept in prototypes developed on our mobile Niccimon platform [1].

2 Related work

Only a few attempts can be found to establish a standard for the definition of POIs, like for instance POIX [2] or NVML [3] and recently OpenLS [4]. POIX and NVML both define POIs and/or route information usable in a city information system, without taking into account context-awareness. However, these “standards” provide first insights and stimuli towards a structured POI data model. OpenLS on the contrary defines services and data types for location services. On this behalf it provides a data type for POIs, which only reflects basic features, e. g., position and time. With regards to context management, the project Nexus [5] works in the field of federating context information and context reasoning, whereas we aim at the modeling of context-aware information of POIs. A first simple example of context-aware interactive objects is given by the Stick-e Notes [6] project, where a single or compound context condition managed

the invocation of a note. However, this only a first basic approach, our *x*POIs supports more complex context situations and aims at a dynamic context-aware presentation. Today, we do not see any comprehensive modeling of context-aware POIs. In most cases more or less hard-wired within the application. All POIs are known beforehand, defined in an application specific data model, and presented in a uniform manner. No dynamic POI exchange or context-aware visualisation is integrated in the POI concepts.

3 The general concept of context-aware POIs

It is the goal of the *x*POI concept to provide self-describing “all-in-one” *x*POIs that carry sufficient information to be processed, managed, and presented according to the users context. This requires a suitable data model to specify and structure the needed information. With regard to the design of the data model, the two central aspects are: presentation and context-awareness. Consequently, a suitable representation of the *x*POI’s visualisation is needed. To achieve as much platform-independence and standard conformity as possible, the visualisation representation should follow widely accepted presentation standards. In order to reflect context-awareness, the *x*POI data model needs to contain the information about the context situation to which the *x*POI is “sensitive”/“aware”. The goal to dynamically adapt the visualisation of a *x*POI necessitates a mechanisms that correlates a context situation with the visualisation presentation, which defines a resulting adaption. Since we also aim to exchange *x*POIs between applications, we further need a suitable transport/exchange format; preferably platform-independent. Finally, a mechanism that is able to process *x*POIs and use *x*POIs together with context information is needed. Our approach to meet these requirements in the *x*POI concept is presented in the following.

4 Design of *x*POIs, context, and *x*POI engine

Based on the general concepts and requirements, we present in this section the design of the main components of the *x*POI concept in more detail: the design of the *x*POI data model, the context data model, and our *x*POI engine which analyses and processes the *x*POIs and the context to generate a context-based *x*POI visualisation and interaction.

4.1 The x POI data model

Our x POI data model is separated into five main sections each responsible for one distinct aspect: identification, management, visualisation, context-awareness, and messaging.

Identification: The *identification* section contains the information needed to identify an x POI. It comprises an unique x POI id, one or more types or categories of the x POI, an action entry to distinguish an initial x POI distribution from an update or delete call, and last, but most important, the actual position of the x POI.

Management: This section contains information for the later *management* of x POIs. Issues addressed are, information concerning security aspects like access rights or the x POI creator, together with the temporal validity, update version, and history of the x POI.

Visualisation: In the *visualisation* section, the possible visualisation forms of the x POI are defined. The visualisation can range from simple text over 2D graphics to auditory and haptic representations. To restrain complexity, we concentrate our work on a 2D representation of the x POIs. Generally, there are two ways to present an x POI: By a bitmap or by a vector object. For the context-aware presentation we apply suitable transformations on both the vector definition and the bitmap definition. For example, in the case of a 2D presentation, the result of this transformation is a SVG object which is later used by the surrounding application to present the x POI on a map.

Context-awareness: The mentioned transformation of the visualisation information is the result of the execution of one of the context-aware rules, defined in the *context-awareness* section. We use Event-Condition-Action (ECA) rules known from, e. g., active databases [7], to define the situation in which a rule is to be executed. Based on the overall rule characteristics, a user may cause an *event*, e. g., by selecting an x POI on the map or by changing the context situation, and/or a specific context situation/*condition* is given, e. g., the user reached a distinct position, and an *action* is performed. Driven by the requirements of our application domain, we restrict the set of supported user **events** to define these rules at this stage of the project to on the one hand “POI-Pressed, POI-Released, and POI-Clicked” for direct user interaction with the x POI and on the

other hand indirect updates on the supported context represented by variables. To specify our **conditions** the supported variables for context like position, position quality, time, date, velocity, role and stress factor are combined with operators and elements of the context domains. The following operators are supported:

conditional: AND, OR, and NOT
relational: EQUAL, LESS, and GREATER
spatial: DISTANCE and INSIDE
constructors: POINT, RECT, TIME, and DATE

In later versions, we will add more operators/context information and examine the usage of different formalisms to specify the context-aware rules. Nevertheless the use of ECA rules is sufficient to demonstrate the usage and advantages of context-aware POIs. Today we support two types of **actions**. First of all, an XSLT transformation can be applied on an appropriate visualisation information. The result of this transformation is the mentioned SVG presentation of the *x*POI. Another action could be the usage of the messaging mechanism to send a message to the *x*POI creator realising context-aware interaction. The following example, in which a *x*POI changes its presentation, if the user move's into a defined area at a specific day, represents a correct ECA rule in the *x*POI-concept, showing only a meta statement for the action part.

Event: UPDATE(*position_{now}*) OR UPDATE(*date_{now}*)
Condition: INSIDE(*position_{now}*, RECT(POINT(0,0), POINT(9,9)))
AND
date_{now} EQUALS DATE(2005,6,21)
Action: *use visualisation form 3*

Messaging: In the section *messaging*, we define information that is needed if a context-aware rule results in a messaging action instead of a visualisation transformation, enabling interprocess communication. Due to the required platform-independence, information regarding the messaging channels like TCP/IP sockets or other mechanisms are provided in this section.

The goal to integrate context-awareness in the POI presentation and foster the exchange and reusability of POIs requires a uniform but flexible *x*POI definition. Therefore, we utilise XML and XML-Schema to specify, express and exchange our *x*POIs.

4.2 Context

In order to use context information information such as the position of the user, the position quality, time, date, velocity, role, and stress factor of the user, we developed a data model for context. It incorporates additional meta data like position information, time related factors, creator, trustworthiness etc and is not described any further. Nevertheless, due to the modular design of the x POI engine (see next section), any other context data model could be used. In addition to the aforementioned context information, which can be used in the definition of the x POIs, we use one additional context element, the *event horizon*. The event horizon describes the area in the real world, which the user is able to comprehend at the moment. This special context element is used during the processing of x POIs.

4.3 Processing of x POIs

In order to utilise x POIs, a mechanism to analyse and process these x POIs is one of the key factors. We propose a so called “ x POI engine” depicted in Figure 1, containing all the necessary functionality.

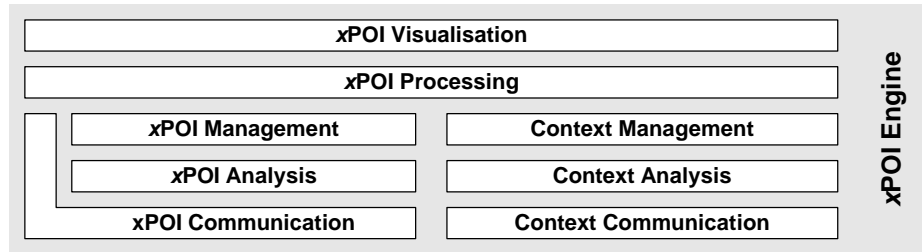


Fig. 1. The x POI engine, processing x POIs

Its task is to be integrated into the later mobile information system, to receive the x POIs and context information, process both, and either calculate the x POIs SVG representations or trigger the context-aware interaction through informing the x POI originator. Figure 1 shows the layers of the x POI engine in which x POIs and context information are integrated, preprocessed, and managed. Both context information and x POIs are integrated into the x POI-engine via the communication layer. Afterwards x POIs as well as the context are analysed and transformed into an internal object representation. Context information and x POIs are then transferred to their

respective management layer, where they are stored “persistently”. Depending on the event (an new x POI has been integrated or an old one has been updated, an user interaction occurred, or the context situation has changed) the affected x POIs and context information are enquired by the processing layer from both management layers. Here the context information *event horizon* is used to extract only those x POIs from the management layer that are of interest in the given situation. After processing the involved x POIs either their new SVG representation is transferred to the visualisation layer and from there to the surrounding system or in case of a communication action the communication layer is used to propagate the information to the x POI originator.

5 Using x POIs in mobile information systems

To show the practical usage of our x POI concept, we integrated the concept into our mobile, multimedia, and location-aware Niccimon platform [1]. Through this, applications that use the Niccimon platform as system foundation are able to utilise the x POI concept. The proposed x POI engine is implemented and integrated in the Niccimon mediator as an additional horizontal module.

The first project to integrate the new x POI concept was our mobiDENK demonstrator [8]. MobiDENK represents a typical guide system, displaying the position of the user and POIs in his or her surrounding on a map, providing location-based information. We are currently testing the new x POIs by addressing proximity awareness, user interests, and time, to adapt their visualisation. Furthermore, we will integrate the x POI concept in our project, Sightseeing4U [9], which provides personalised sightseeing tours on mobile devices. Here, the x POIs will be adapted according to the user’s interests regarding architecture or landscaping.

6 Conclusion

In this paper, we introduced a “standard” x POI data model that enables the creation, exchange, and processing of context-aware points of interest in mobile, location- and context-aware applications. These x POIs each carry the information to be identified, managed, and, most important, be visualised dependent on the users current context situation. We not only presented the data model for x POIs but

also the processing architecture in which they are embedded. The concept invites new interaction models in mobile computing. Based on this “context sensitive interaction” users can interact with the system by changing their context, e. g., walking around, changing their role from tourist to business person, and the like. Due to the platform-independent exchange format of *x*POIs, applications can easily exchange and integrate context-aware POIs. When thinking of teenagers and their struggle for individuality the exchange of individualised *x*POIs among themselves could prove to be an interesting application scenario. Nevertheless there are some open issues like for instance the diffuse definition of context or the definition of “useful” rules together with a sound visualisation transformation. But, the implementation and application of the *x*POI concept so far is promising. In a world of evolving mobile and context-aware applications, the flexible usage and exchange of context-aware points of interest clearly is an important next step.

References

1. Baldzer, J., Boll, S., Klante, P., Krösche, J., et al.: Location-Aware Mobile Multimedia Applications on the Niccimon Platform. In: Informationssysteme für mobile Anwendungen (IMA'04), Brunswick, Germany (2004)
2. Kanemitsu, H., Kamada, T.: POIX: Point Of Interest eXchange language, version 2.0, document revision 1. W3C NOTE (1999)
3. Sekiguchi, M., Takayama, K., Naito, H., Maeda, Y., Horai, H., Toriumi, M.: NaVigation Markup Language (NVML). W3C NOTE (1999)
4. Bychowski, T., Williams, J., Niedzwiadek, H., Bishr, Y., et al.: OpenGIS Location Services (OpenLS). OpenGIS Implementation Specification OGC 03-006r3 (2004)
5. Hohl, F., Kubach, U., Leonhardi, A., Rothermel, K., Schwehm, M.: Next century challenges: Nexus - an open global infrastructure for spatial-aware applications. In: 5th Mobile Computing and Networking, Seattle, USA (1999) 249–255
6. Pascoe, J.: The stick-e note architecture: Extending th interface beyond the user. In: IUI '97: Intelligent User Interfaces, Orlando, Florida, USA (1997) 261–264
7. Gatzju, S., Geppert, A., Dittrich, K.R.: The samos active dbms prototype. Technical Report 94.16, IFI (1994)
8. Krösche, J., Boll, S., Baldzer, J.: MobiDENK – Mobile Multimedia in Monument Conservation. IEEE MultiMedia **11** (2004) 72–77
9. Boll, S., Krösche, J., Scherp, A.: Personalized Mobile Multimedia meets Location-Based Services. In: INFORMATIK 2004 - Informatik verbindet, Band 2 “Multimedia-Informationssysteme”. Volume 51 of LNI., Ulm, Germany (2004)