

# Introducing IYOUIT

Sebastian Boehm<sup>1</sup>, Johan Koolwaaij<sup>2</sup>, Marko Luther<sup>1</sup>, Bertrand Souville<sup>1</sup>  
Matthias Wagner<sup>1</sup>, and Martin Wibbels<sup>2</sup>

<sup>1</sup>DoCoMo Euro-Labs, Landsbergerstr. 312, 80687 Munich, Germany

<sup>2</sup>Telematica Instituut, Brouwerijstraat 1, 7523 XC Enschede, The Netherlands

**Abstract.** We present IYOUIT, a prototype service to pioneer a context-aware mobile digital lifestyle and its reflection on the Web. The application is based on a distributed infrastructure that incorporates Semantic Web technologies in several places to derive qualitative interpretations of a user's digital traces in the real world. Networked components map quantitative sensor data to qualitative abstractions represented in formal ontologies. Subsequent classification processes combine these with formalized domain knowledge to derive meaningful interpretations and to recognize exceptional events in context histories. The application is made available on Nokia Series-60 phones and designed to seamlessly run 24/7.

## 1. Introduction

In this paper we introduce IYOUIT<sup>1</sup>, a mobile application that allows users to automatically collect so-called context information centered on places they visit and people they meet. Context, in a technical sense, is regarded as any piece of information that can be recognized and further processed to adapt the behavior of the application according to a given set of constraints [1]. The application aims at making it easy to automatically collect such data with a standard phone and facilitates an instant and light-hearted sharing of personal experiences within communities.

All data collected by IYOUIT is aggregated into a wealth of context information and made accessible to users on the Web and on a mobile client. For selected context sources, value is added through the transformation of quantitative context information into qualitative statements about a user's given situation. By hooking up to Web 2.0 services like Flickr<sup>2</sup> and Twitter<sup>3</sup>, the application allows for sharing personal context with others online. Sharing can be instant, by posting single data items, or through the aggregated contextual experience in potentially lifelong online blogs.

In IYOUIT and its underlying component framework, Semantic Web technology is used in several places to implement key features and to seamlessly connect the application to different services on the Web. IYOUIT is the result of our long-year efforts to leverage the use of context information in mobile applications and has recently

---

<sup>1</sup> <http://www.iyouit.eu>

<sup>2</sup> <http://www.flickr.com>

<sup>3</sup> <http://twitter.com>

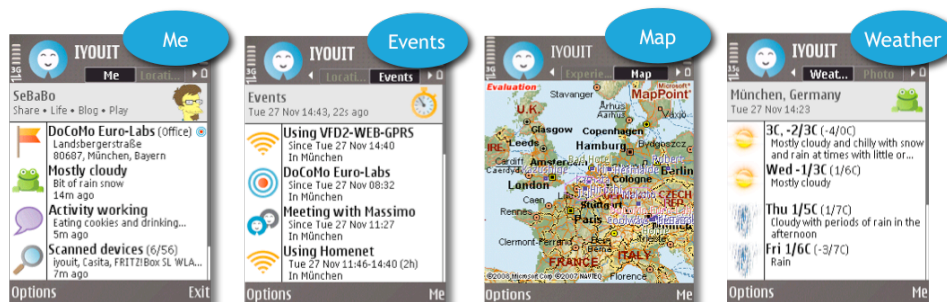


Figure 1: The IYOUIT Mobile Client

been rolled out of our labs as a prototype service for free public use. The remainder of this paper is organized as follows: in Section 2 we give a first conceptual overview of IYOUIT and explain its target application domains, which correspond to our general research objectives. In Section 3 we show where Semantic Web technology is embedded into our systems. We do so in concretely discussing selected implementations of the underlying component framework. Section 4 illustrates the practical use of IYOUIT and highlights features that actually reveal semantics to the user. We summarize and conclude with a view on the current IYOUIT user community.

## 2. A First Look at IYOUIT

IYOUIT has been designed and developed as a “living lab” for our research on mobile community services, context awareness and the smart fusion of both towards the Semantic Web. The idea has always been to spin-off and implement early ideas from research into this living lab for feedback and evaluation within the user community. IYOUIT is the reference implementation of our underlying component framework and comes as a mobile client (a first impression is given in Figure 1) as well as a Web portal and integrates popular 3<sup>rd</sup> party Web 2.0 services like Flickr and Twitter.

Since recently, IYOUIT is released to the public as a free service with a wide set of implemented features that have been streamlined towards four complementary, but not mutually exclusive, application domains *Share*, *Life*, *Blog* and *Play*. In the spirit of our living lab approach, all features that are showing in the application are fully functional, yet it remains to be seen how well they will be accepted and used by the IYOUIT user community.

In this section we give a first and high-level overview of the IYOUIT application domains as well as the implemented features. We also introduce the underlying context management framework.

### 2.1 Share, Life, Blog, Play

**Share** (*community-based context sharing*): application features of the IYOUIT target domain *Share* are concerned with the possible synergies of context-awareness and social networking services. The work builds on results [2] in social networking en-

hanced through ontology-based reasoning for communities and focuses on the analysis of personal context histories and established relationships to identify possible social network extensions. The social network of users with qualified social relationships is represented using formal ontologies. Through ontology-based reasoning, data consistency can be ensured and additional relationships can be deduced to complement the users' circle of friends as a social portfolio.

**Life** (*life support through context-aware guidance*): IYOUIT *Life* deals with analyzing user-generated content, such as tags that are manually assigned to photos, and its relation to context over time and space. In the spirit of the Web 2.0, the goal is to extract information from the “wisdom of crowds” through aggregation in the geographic space. As starting point we apply clustering to user-generated tags, which hold location information, to determine regions of interest on maps. These tag clusters – characterized by population, density, position and range – can be used to guide IYOUIT users in their surrounding area. For data sources, IYOUIT *Life* builds on the services Flickr and Flagr<sup>4</sup> where geo-tagging already enjoys increasing popularity.

**Blog** (*enhanced contextual blogging*): IYOUIT *Blog* is concerned with enhanced automatic blogging capabilities based on our related work in ontology-based reasoning [3] and a more specialized reasoning [4] to recognize and manage complex contextual events. As most collected context data in IYOUIT is of quantitative nature, abstraction methods and context ontologies have been introduced to deal with context at a higher level. At the level of these context ontologies, complex conceptual dependencies between context elements are introduced to enrich contextual descriptions and to implement classification-based reasoning about the user's situation, e.g. to describe user places as conceptual abstractions from exact locations (“Office”, “Home” or “Business Place”).

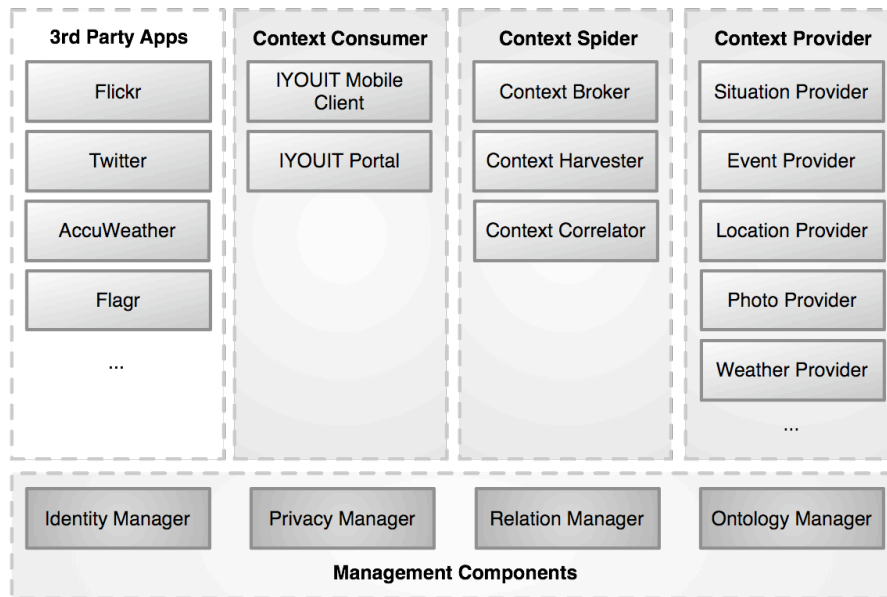
**Play** (*playful experience of context-awareness in games*): in IYOUIT *Play* we focus on low-level and quantitative aspects of context gathering, management and similarity detection. Work is concerned with detecting specific context constellations based on low-level context features and describing these constellations in both, a formal and a human readable format. Context histories are observed to discover homogeneous time segments in a higher-dimensional context space and to detect strong correlations between different context dimensions. From the four IYOUIT target domains, Play is at present and by design the one that is the least grounded in Semantic Web technology. It is therefore not further addressed here.

## 2.2 Context Management

IYOUIT is based on its own Context Management Framework (CMF) to host various services and data sources. Framework components, for instance, track the positions of users via GPS and cell tower information to identify frequently visited places over time. Further context sources include the whereabouts of buddies, scanned Bluetooth and WLAN beacons, local weather, photos, sounds, observed products, books or messages.

---

<sup>4</sup> <http://www.flagr.com>



**Figure 2:** IYOUIT Context Management Framework

Through distributed CMF components, the gathering of context data can be implemented in a flexible way, to reason about the combination of various context streams. In the following we give an overview of all essential framework components (cf. Figure 2).

**Management Components:** the management components are the foundational CMF building blocks. They, for instance, ensure a secure authentication of entities (Identity Manager), enable access control mechanisms to prevent the uncontrolled disclosure of sensitive information (Privacy Manager, see [5]), allow for the usage of domain specific knowledge formalized within a set of core ontologies (Ontology Manager, cf. Section 3.1), or provide means to represent and reason about the social network of users (Relation Manager, cf. Section 3.2).

**Context Spiders:** the Context Broker, Context Harvester and the Context Correlator form the group of Context Spiders and provide means for easy context lookup, gathering and alignment across components. The Context Broker maintains a complete repository of all registered components and their public schemata to discover appropriate services and to allocate queries to the respective component. The Context Harvester retrieves context data across multiple components independent of where the information is stored physically. The Context Correlator aligns distinct context streams based on their temporal intersection and identifies significant correlations over time.

**Context Providers:** Context Providers (CP) lie at the very heart of the CMF. They encapsulate the basic context data sources at a quantitative level but can also implement aggregations and abstractions to a qualitative level (cf. Section 3.3 and 3.4). Each CP has been designed to realize domain specific services based on the underly-

ing core infrastructure. To do so, a CP first gathers a certain type of information (e.g. the user's cell-id) from a sensor (for instance the mobile handset) or another CP to further process and combine this information with other context data.

**Context Consumers & 3rd Party Applications:** the IYOUIT mobile client, the IYOUIT Web portal as well as 3rd party integrations are leveraging CMF through the model of Context Consumers. Access to the core components is implemented through the Privacy and Identity Manager. As 3rd party applications, currently, Flickr, Twitter and Google Earth<sup>5</sup> are supported.

### 3. Embedded Semantics

The main objective of our Context Management Framework is to abstract from raw sensor data to eventually gain qualitative information about a user in a given situation. We assume that the meaningful interpretation of context is only feasible at a qualitative level, based on aggregated context data. To determine a common vocabulary for a unified interpretation of qualitative context among CMF components we designed a set of specific context ontologies formulated in a decidable fragment of the Web Ontology Language (OWL) [6]. Each Context Provider is responsible to link the quantitative values contained in context elements to qualitative values expressed using this vocabulary. In addition, a Context Provider might also interface with OWL reasoning engines to derive even higher-level of abstractions through the classification of sets of qualitative values using standard Description Logic [7] techniques.

In the following, we give insights into how semantic technology is applied in IYOUIT and highlight the possible alignment of context data to well formed ontologies in a practical application.

#### 3.1 Ontology Manager

To make use of ontology based reasoning mechanisms, we designed a set of core ontologies for distinct application domains including social relationships, location records and weather conditions, amongst others. A total number of twelve interrelated component ontologies defining more than 300 concepts, 200 properties and 300 individuals have been defined. Due to near real-time requirements of most applied reasoning processes, we concentrated on the modeling of highly efficient yet expressive enough ontologies rather than utilizing already existing and widely used ontologies such as DOLCE<sup>6</sup>. Furthermore, these ontologies are not used as the main representation format for all aspects of context, since ontologies are generally weak in handling large amounts of data efficiently [8]. Instead, only distinct higher-level data elements are annotated with ontology references, making them available for further ontology reasoning. This way, the overall scalability is not affected, while at the same time valuable reasoning results can be achieved.

---

<sup>5</sup> <http://earth.google.com>

<sup>6</sup> <http://www.loa-cnr.it/DOLCE.html>

The Ontology Manager provides a gateway within the CMF to simplify the access to OWL ontologies. Given an OWL ontology, it allocates a reasoner resource and instructs it to retrieve the corresponding set of axioms. As foreseen in the OWL 2 draft specification [6], the access to an ontology is accomplished through a physical URI that is given by a mapping from its identifying logical URI. This configurable ontology mapping, the parsing of the concrete OWL syntax and the interfacing with various DL reasoners (at the time of writing we use Pellet [9], FaCT++ [10] and RacerPro [11]) is delegated to a semantic middleware (in our case the OWL API [12]).

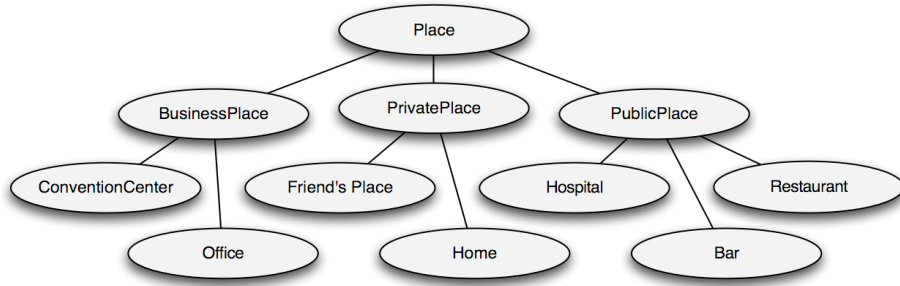
After having configured the Ontology Manager appropriately, a CMF component can request structural information about named concepts, properties and individuals. Among the supported queries are requests to retrieve (parts of) the concept and property hierarchy, the (direct) types of individuals, the (direct) individuals of concepts and the relations that hold between two individuals. Each of those requests might also refer to implicit knowledge, which is why a reasoner has to be involved to ensure the completeness of the returned results.

### 3.2 Relation Manager

All of IYOUIT's community services rely on social networking. Users can establish defined relationships amongst each other, e.g. "friend", "colleague" or "husband", to build social networks and share context data through them. IYOUIT, for instance, supports the concept of buddy lists on its mobile client where users can instantly look up what their friends are doing. Likewise, photos, sounds and other context elements can be shared with others on the Web.

Social networking is implemented through the Relation Manager that exposes a specialized API to allow restricted access to sensitive social context data. The Relation Manager stores social networks as OWL ontologies and enables DL-based reasoning to classify and to combine explicit, user-defined facts about social relationships with the world knowledge encoded for network completion [2]. In short, OWL-based reasoning is crucial for two reasons: for one, the consistency of the provided data can be maintained by dismissing any contradicting or illegitimate definitions; secondly, the applied knowledge discovery techniques can complement the social network of users in deducing implicit relationships.

The entire set of all relationships, including user-defined as well as deduced relationships, is stored within a relational database. The inference process is triggered whenever a relationship has been approved or in case an existing relationship has been removed. All relationships are then transmitted to the inference engine, with the reasoning results again being stored in the database. As a result, the underlying knowledge base does not need to be involved in most cases, because the majority of requests from other CMF components are concerned with retrieving rather than modifying data.



**Figure 3:** Snapshot of the Place Ontology

The Relation Manager is used to implement access control policies for privacy protection in IYOUIT: not only is an established relationship between users essentially required to share data, privacy policies can also be defined along relationships to determine the level of access to personal context. Such access policies can be specified for individuals (“my friend Robert”) or types of individuals (“all my friends”) and are naturally bound to the underlying social network. The level of access detail is defined per context category, e.g. “show address information only to the level of the city name”.

For many common relationships and access policies IYOUIT provides a predefined social ontology and access control directives. To this end, we have modeled the social ontology, in which a hierarchy of more than 50 social relationships has been described. This way sophisticated privacy directives can be expressed to, for instance, specify that colleagues should only know the city of his current location whereas family members may have access to the entire record.

### 3.3 Location Provider

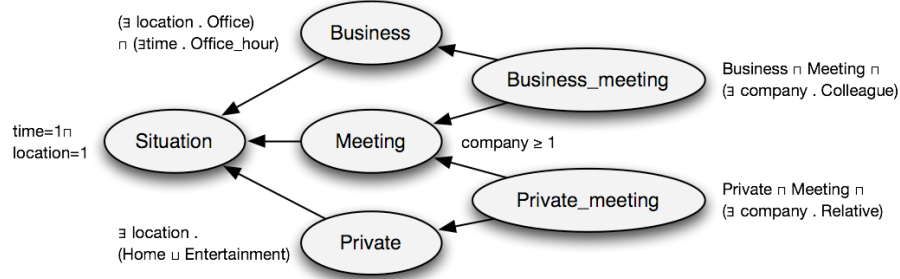
The main task of the Location Provider is to resolve given location estimations into actual address records, to store location traces and to deduce frequently visited places.

To abstract from exact positions to conceptual places, the Location Provider identifies significant location records of a user by applying profound statistical learning and clustering methods to historic location data [13]. Once established, a place is presented to the user to name and typify it by selecting an appropriate concept from the place ontology, which includes descriptions like “Office”, “Home” or “Business Place” (see Figure 3). Staying in a place is from now on recognized automatically by IYOUIT, resulting in qualitative location characteristics also shared among buddies.

### 3.4 Weather Provider

The Weather Provider enriches a given location record with prevalent meteorological data such as the actual temperature or wind speed. It employs two types of automatic abstractions that link qualitative to quantitative information.

Simple static mappings assign values within certain intervals to the corresponding abstract descriptions. One example is the widely used Beaufort scale [14], which



**Figure 4:** Snapshot of the Situation Ontology

maps wind speeds to 13 qualitative wind conditions, e.g. “light air” or “gentle breeze”, that are in turn further characterized within the weather ontology.

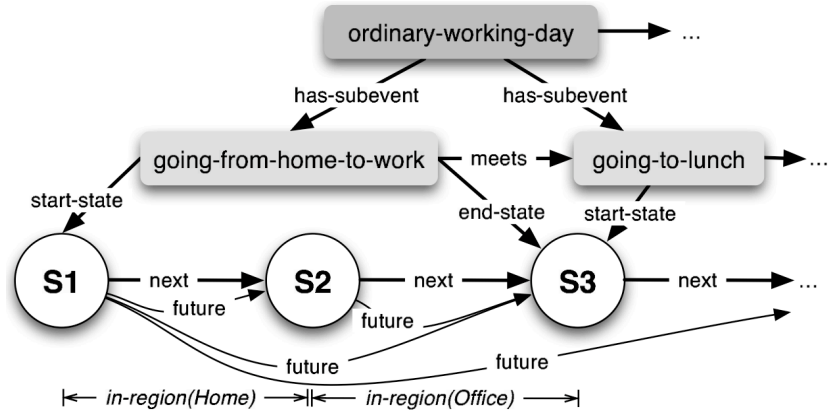
To categorize quantitative temperature values meaningfully, more complex mappings are needed that take additional context into account. A static mapping as the one described before, might otherwise categorize a given temperature of 5°C as “cool”, even though it might have been recorded on an exceptionally warm day somewhere in the north during winter time. Instead, the Weather Provider considers several types of context to dynamically derive a sensible mapping for temperatures and precipitations. For this, historical weather records that provide monthly minimal and maximal mean temperatures and precipitation values are used. In addition, the location, the day of the year and the hour of the day are taken into account to derive a rational categorization of the actual qualitative values such as “low”, “moderate”, “warm”, etc. Note that this mapping is slightly more evolved than the dynamic mapping of places accomplished within the Location Provider as it takes multiple types of context into account and does not involve user interaction for the final semantic categorization.

Mapping of large sets of meteorological data to qualitative abstractions allows for rating an overall weather condition, represented as an individual and linked via object properties to corresponding meteorological abstractions by a standard DL classification process. We defined categories for “bad”, “fair”, “good” and “splendid” weather conditions based on the ratings for individual weather attributes like temperature, wind, pressure or precipitation and a qualitative health index. This health index is itself derived by classification and expresses to what extent the current weather situation may cause aches and pains. Weather health issues such as chronic pains, aching bones or migraine are again recognized during the classification, based on formalized background knowledge. For instance, an axiom formalizes that weather conditions with low pressure, the passage of a warm front, high temperatures and humidity often cause migraines.

### 3.5 Situation Provider

The Situation Provider computes an abstract characterization of a user's situation by applying DL classification on several context pieces gathered by the Context Harvester from multiple Context Providers [3]. Abstract situation concepts like “Business Meeting” are formulated w.r.t. the vocabulary of the respective component ontologies





**Figure 5:** Example Event of the CMF Event Provider

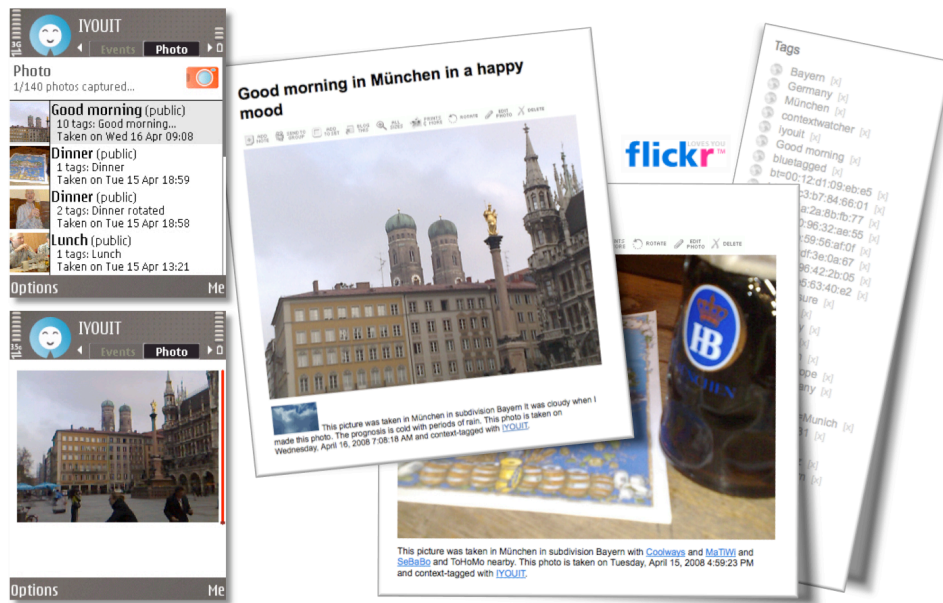
(cf. Figure 4). Each situation individual is assembled of a set of entities representing qualitative context information such as the location (e.g., office), the time (e.g., afternoon) and people in proximity (e.g., friends). Finally, the classification result given by the computed direct individual types is established as a new (derived) context.

### 3.6 Event Provider

The reasoning within the Situation Provider to classify user context can be seen as static in time as it observes the situation of a user only for one given point in time. To reason about complete context histories over longer periods, specialized reasoning is needed beyond DL-based classifications to detect the line of key situations over time.

The Event Provider is targeting at such a detection of key events, which are defined as temporal and spatial constellations of significant situations in context histories. The primary goal is to use detected events for an optimized composition of on-line blogs to meaningfully structure blog entries according to their contextual significance. In [3] we propose an approach based on RacerPro and the semantic query language nRQL [15], which is currently being implemented within the CMF Event Provider. The work exploits complex location concepts, e.g. “my father’s house”, quantitative concepts of time as well as the linkage of both with other significant context elements. Subsequently, spatio-temporal events like “leaving the home” or “returning from a vacation” can be expressed to indicate a qualitative spatio-temporal change in the user’s state or situation.

Figure 5 gives examples of events as they can be expressed in our model. An event is defined as a time interval having a start state and an end state. It either describes a constancy that holds between states, e.g., like “staying at home”, or a certain change that happens, e.g. “going from home to work”. Events may naturally depend on each other and can be organized hierarchically. In the model we further distinguish between events as being homogeneous, complex and high-level to allow for the flexible design of large knowledge bases and their reuse in different application domains.



**Figure 6:** Photo Share

## 4. IYOUIT in Everyday Life

In this section we run through three actual use cases of IYOUIT to illustrate how implemented application features can be bundled to realize the IYOUIT target domains of Share, Life and Blog.

All examples, descriptions and screenshots are taken from the actual IYOUIT service as available today. The mobile client is organized in application tabs, where each tab either displays a certain type of context information (e.g., local weather reports in the Weather tab) or accumulates various pieces of information in a context overview. We encourage all readers who carry a Nokia Series-60 phone to visit our Web site, download IYOUIT and experience the described scenarios in real life.

### 4.1 Photo Share

Photo Share is the first use case presented in here and reflects the domain of IYOUIT Share. As shown in Figure 6, the mobile client provides access to the phone camera and allows the user to take photos while on the go. Once a picture is captured, IYOUIT automatically compiles all context information available at the given moment and proactively adds it to the photo. Photos are then directly published on Flickr and can be shared with others online.

On Flickr, the automatically added contextual data is made available in photo tags. Semantically enriched bits of information are added in complementary tags, the photo caption or the photo description. Also showing is the derived situation in terms of



**Figure 7:** Buddy Map

place, nearby buddies and the prevalent weather condition, qualified through the underlying ontology concepts.

The comprehensive tagging capabilities of Photo Share aim at maximizing the use of Flickr for IYOUIT users and allow photos to be organized along the given tags with ease. Photo albums can, for instance, be easily queried for all pictures matching the search for “Pics that I took abroad during lunch on a sunny day?” or “Pics that I took during a private trip to a far destination with my family?”.

#### 4.2 Buddy Map

As mentioned above, social networking and buddy-centric context sharing are key features in IYOUIT. Figure 7 depicts a strong use case that is created in combining the contextual knowledge centered on locations and places with social networking.

The use case of Buddy Map aims at simplifying the daily life of users through keeping track of frequently visited places. Those places add significantly to the semantic value of a user’s context, for instance in detecting complex events, but can also be used straightforwardly for basic features: the appearance of buddies in places can trigger small alerts, views to context data can be adjusted based on the type of place, or settings of the phone can be automatically switched based on rules bound to places. As an example, an ad-hoc meeting can be quickly setup in the “Office” place based on the observed fact that all “co-workers” are present in that place. During the actual meeting situation (where again place information is central to detect it) phones can be automatically muted.

The examples in Figure 7 show a view to the IYOUIT buddy list that is grouping people according to the place they are currently staying in. The right part of the figure depicts actual places on a map revealing their approximated location and size as well as some additional attributes.

#### 4.3 Life Blog

Changes in the context of a user can be hints for significant changes of a current situation and experience. Users may wish to communicate such changes to others in subtle



Figure 8: Life Blog

ways, e.g. instead of calling the family at home to say “I just left the office and will be home soon” a small automatic note or a change in the presence settings will indicate that you are now commuting and back at home soon.

This is the essential idea of our “Life Blog” use case as depicted in Figure 8, where changes in the personal context can be automatically posted to others as micro-blog messages. This is done automatically through the IYOUIT mobile client or on the Web. In the example, within the buddy list, recent changes are indicated for six different users in their context data on location, local weather and observed objects. Complementary, and as also shown in Figure 8, similar contextual notes are posted online on the IYOUIT Web portal and aligned on a map. This way it becomes easy for an observer to follow buddies and to get an idea of what they are up to. In addition to automatic, contextual postings users can submit small text notes to their buddies or the whole user community. These notes are enriched with contextual data and posted to Twitter. In the example, place and location data is automatically added as the message context together with significant presence attributes.

Context histories and micro-blog postings over longer durations are also compiled into permanent blog entries (not shown in the example) to summarize significant periods of time. Such blog entries can, for instance, summarize a trip based on visited locations, experienced weather, buddies in company and the captured photos. Furthermore, the detection of complex events within the observed context history can be used to organize such blogs.

## 5. Summary and Conclusion

We have introduced IYOUIT, a mobile application to enrich the digital life of people on the go. The application is available through our Web site and can be used on any standard Nokia Series-60 phone. It implements many ideas from the fields of context awareness and community-based sharing while connecting to the established Web 2.0 communities of Flickr and Twitter. As a stable research prototype from our labs, it has been released to the public in June 2008.

We are aware of competing mobile applications like ZoneTag<sup>7</sup>, Merkitys-Meaning<sup>8</sup> or Shozu<sup>9</sup> that partially follow similar aims but seem to fall short in the completeness of their key features. A more detailed comparison with these applications is discussed elsewhere [5]. To the best of our knowledge, no applications currently exist that try to leverage Semantic Web technology in similar ways for the mobile domain.

Semantic Web technology is applied in IYOUIT core components, in particular for qualitative context abstraction and reasoning. We have put effort in optimizing our OWL ontologies and focused on efficiently integrating inference techniques based on results gained from extensive evaluation studies [8]. Work in progress is concerned with further extensions to the classification-based reasoning on user situations towards complex events along the spatio-temporal dimensions of context data. However, we also plan to model and publish meta-data of selected public context items with standard vocabularies such as FOAF<sup>10</sup> and SIOC<sup>11</sup> to interlink with other services on the Semantic Web.

With IYOUIT we are taking an agile approach for a rapid development and early release of application features. In the current version, features are manifold and purposely put quite broad. We plan to continuously rectify features, add further functionality or also remove some, based on their popular use as well as the feedback and demands of our users. Albeit early in its development and release, the YOUTIT user community is already substantial: to date we support more than 400 users. Amongst other context data items, they have generated more than 250.000 location measurements, took 2.500+ photos, visited over 4.000 cities in 30 different countries where about 12.000 local weather reports were received.

**Acknowledgements.** We would like to thank all IYOUIT users, especially our friends and co-workers at DoCoMo and Telematica Instituut, for their great feedback and support. We are grateful to our colleagues at DoCoMo i-mode Europe for fruitful discussions on the potential of IYOUIT, compelling real-world use cases and the competitiveness of IYOUIT in the fast-paced world of Web 2.0. Finally, special thanks go to Michael Wessel of Racer Systems for the support and efforts in developing and implementing the spatio-temporal event recognition in context histories.

---

<sup>7</sup> <http://zonetag.research.yahoo.com>

<sup>8</sup> <http://meaning.3xi.org>

<sup>9</sup> <http://www.shozu.com>

<sup>10</sup> <http://www.foaf-project.org/>

<sup>11</sup> <http://sioc-project.org/>

## References

1. A. K. Dey, "Understanding and using context," *Personal and Ubiquitous Computing Journal*, vol. 5, no. 16, pp. 4–7, 2001.
2. S. Böhm, M. Luther, and M. Wagner, "Smarter groups – reasoning on qualitative information from your desktop," in *Proc. of the Workshop on The Semantic Desktop*, 276–280, November 2005.
3. M. Luther, Y. Fukazawa, M. Wagner, and S. Kurakake, "Situational reasoning for task-oriented mobile service recommendation," *The Knowledge Engineering Review*, vol. 23, pp. 7–19, March 2008.
4. M. Wessel, M. Luther, and M. Wagner, "The difference a day makes – recognizing important events in daily context logs," in *Proc. of the 2nd Workshop on Context and Ontologies in conj. with CONTEXT'07*, August, 2007.
5. S. Böhm, J. Koolwaaij and M. Luther, "Share Whatever You Like," in *Proc. of the 1st Int. DisCoTec Workshop on Context-aware Adaptation Mechanisms for Pervasive and Ubiquitous Services (CAMPUS'08)*, June 2008.
6. B. Motik, P. Patel-Schneider, and I. Horrocks, "OWL 2 Web Ontology Language: Structural specification and functional-style syntax," W3C Working Draft, World Wide Web Consortium, April 2008.
7. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, *The Description Logic Handbook – Theory, Implementation and Applications*. Cambridge University Press, January 2003.
8. T. Weithöner, T. Liebig, M. Luther, S. Böhm, F.W. von Henke, and O. Noppens, "Real-world reasoning with OWL," in *Proc. of the 4th European Semantic Web Conference (ESWC'07)*, pp. 296–310, June 2007.
9. E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, "Pellet: A practical OWL-DL reasoner," *Journal of Web Semantics*, vol. 5, pp. 51–53, June 2007.
10. D. Tsarkov and I. Horrocks, "FaCT++ description logic reasoner: System description," in *Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR'06)*, vol. 4130 of *LNAI*, pp. 292–297, Springer, August 2006.
11. V. Haarslev and R. Möller, "Racer: A core inference engine for the Semantic Web Ontology Language (OWL)," in *Proc. of the 2nd Int. Workshop on Evaluation of Ontology-based Tools (EON'03)*, vol. 87, pp. 27–36, CEUR-WS.org, 2003.
12. M. Horridge, S. Bechhofer, and O. Noppens, "Igniting the OWL 1.1 touch paper: The OWL API," in *Proc. of the 3rd Int. Workshop on OWL Experiences and Directions*, June 2007.
13. P. Nurmi and J. Koolwaaij, "Identifying meaningful locations," in *Proc. of the 3rd Int. Conf. on Mobile and Ubiquitous Systems: Networks and Services (MobiQutous'06)*, IEEE Computer Society, July 2006.
14. S. Huler, *Defining the Wind: The Beaufort Scale*. New York: Crown, 2004.
15. M. Wessel and R. Möller, "A High Performance Semantic Web Query Answering Engine," in *Proc. Int. Workshop on Description Logics (DL '05)*, July 2005.