ABSTRACT
In a world of global networking, the increasing amount of heterogeneous information, available through a variety of channels, has made it difficult for users to find the information they need in the current situation, at the right level of detail. This is true not only when accessing information from mobile devices, characterized by limited – although growing – resources and by high connection costs, but also when using powerful systems, since the amount of "out-of-context" answers to a given user request may be overwhelming. The knowledge of the context in which the data are going to be used can support the process of focussing on currently useful, personalized information. The activity needed for context-aware information personalization provides material for stimulating research, briefly illustrated in this paper.

1. INTRODUCTION
Information is personalized based on the characteristics (interests, social category, current situation, ...) of an individual. Over time, the basis for personalization has evolved from the notion of a simple profile into the collection of all pieces of information that can be used to characterize the situation of an entity such as a person, a place, or any other relevant object/aspect [11, 27, 2, 8] in the interaction between a user and a system: in one word, the context. A further important trait of effective personalization is that information reshaping is mainly based on implicit data, derived by the system from the external environment or from the users' behaviour.

In the Context-ADDICT project [10] context is hierarchically modelled in terms of observable parameters that have a symbolic internal representation within a context schema (in our research, we use a structure called Context Dimension Tree [9]), and some of which correspond to numerical values gathered from the environment by means of suitable appliances, such as sensors or RFID tags. At run time, the context is "sensed", and then validated, by verifying that the discovered combinations of values are consistent w.r.t. the context schema. This triggers a context-aware behaviour: e.g., the delivery of context-aware data or the actuation of context-aware operations. Note that the context model proposed in Context-ADDICT is completely orthogonal w.r.t. the kind of specific context-aware task, thus can easily be adopted, for instance, to tailor and compose Web services according to the current context, to adapt interfaces, to do context-aware information retrieval or to contextualize non-functional requirements like data quality [6] or similar.

In particular, information personalization via context-aware data tailoring consists of (i) extracting the feasible contexts from a context specification (the Context Dimension Tree), and (ii) conveying to the user the context-dependent views associated with the (possibly multiple) current context(s), thus retaining, from the underlying dataset, only the relevant data for each of such contexts.

The challenges involved in context-aware information personalization provide material for stimulating research, needed to establish the foundations of the personalization process. We consider a general scenario where several classes of users access, through a (set of) application(s), a variety of information, made available by internal or external data sources, from sensors used for monitoring the environment to datasets of any format. The knowledge of the context is used to control the flow of information reaching the users, excluding the information noise. A fundamental aspect of context management is to keep the data manipulation operations used to compute and to present the personalized information transparent to the user and independent of the logical structure of the information source. The whole process of information personalization can be mainly described by the following functionalities:

a) **(Dynamic) Context Management**: the role of the designer is to envisage the possible contexts that will be possibly experimented by the user or the application itself during the system’s life. As already noticed, the activity of context modeling is in principle independent of the process and of the data we want to personalize and whose relationships with the different contexts are to be specified apart.

The requirements of a context-aware system are intrinsically dynamic and can evolve. As a consequence, it is highly probable that the context perspectives anticipated at design time will not be always applicable. Accordingly, the context schema used as the basis of the personalisation process should also be allowed to change over time, along with the design of the related context-aware data or operations. These problems are briefly described in Section 2.
In the next sections, starting from the current investigation on the challenges that await researchers on these topics.

### b) Context-Aware Personalization Management

At each instant of time, the currently active context (possibly more than one) has to be recognized: some of the aspects characterizing the current context can be automatically derived from the environment (such as the location and time of the user, her/his role within the application scenario, or the kind of device used to access the data) and others may require an explicit specification by the user or the application of learning capabilities (for instance, the user’s current topics of interest). Once the context has been recognized, contextualization must be applied to the user queries, and the context-aware data must be delivered to the users and applications. Note that the user interests and preferences may also vary according to the context the user is currently in, for a change in context may change the relative importance of information. **Contextual preferences** are then used to refine the views associated with contexts, by imposing a ranking on the data of a context-aware view and adding the opportunity to dispose of the less interesting portion if needed. This research and the associated challenges are briefly described in Section 3.

### c) Data Access Management

The proliferation of freely-accessible data-intensive websites, as well as initiatives for open-accessible linked data in the Web [7], provide the users with potential sources of precious information. Such heterogeneous, dynamic, possibly mobile data sources must be queried in a context-aware fashion, retrieving only the data that are consistent with the current context. The task itself of recognizing the current context requires “sensing” the context data from the environment before transforming them into the actual values for the context parameters, thus the sensors constitute another possible data source that must be dealt with, this time to derive the current context. More about this will be described in Section 4.

In the next sections, starting from the current investigation on context-based personalization in our group, we will illustrate some of the challenges that await researchers on these topics.

## 2. Dynamic Context Management

A key element of context-aware system design is the representation and management of context and its attributes, along with the relationships between a context and the adaptive aspects of the system and application. In order to define a context-aware system it is in fact necessary to identify and model the aspects that characterize the different contextual situations and how such aspects have an impact of the elements of interest for the application scenario, such as the relevant information, applicable rules and behaviors, available features and services.

Our initial effort has thus been devoted to the definition of a context model, formally presented in [9], able to capture the designer’s specification of the elements characterizing the application scenarios and all possible contexts the user might be acting in. The model is characterized by generality (suited for any scenario), multiple abstraction levels (describing the context perspectives at the correct level of granularity), expressivity (allowing querying, reasoning or constraint specification on the contexts of a given target application) [8, 19]. In particular, the proposed model provides the constructs to design the context schema (or Context Dimension Tree – CDT) of a given scenario by means of a hierarchical structure consisting of i) context dimensions (black nodes), modeling the different perspectives through which the user perceives the application domain and ii) the allowed dimension values (white nodes), i.e. the actualized parameters based on which the context-aware information, behaviors, etc. are to be selected. Each context is then defined as a combination of context elements, that is, \(<\text{dimension-value}>\) pairs. The adoption of a hierarchical organization allows us to employ different abstraction levels to specify and represent contexts. It is also possible to define appropriate constraints on the schema, that prevent some of the combinations of context elements, if meaningless for the current application scenario.

The system/application designer specifies the context model at design-time, identifying the possible contexts the user may find him/herself in at run-time. As an example, consider the tourism domain and a mobile application that offers to tourists with different characteristics (e.g. age, nationality, social category) fresh information about restaurants, hotels and exhibitions related to the place where they are. For instance, \(<\text{user=family, interest_topic=restaurants, season=summer, location=\text{“Paris”}}\>) constitutes a possible context for a family interested in finding information about restaurants in Paris during the summer period.

By focusing the attention on relevant information based on the user’s active context, we developed a methodology for the semi-automatic tailoring of data, starting from the definition of a view associated with each context element, that determines the portion of data deemed interesting when the active context includes such element. The view has been dubbed relevant area, or \(\text{Rel}\), and the methodology, given a context, allows the automatic computation of the relevant data by appropriately combining the relevant areas associated with each one of the involved context elements. At design time, the designer must specify the view associated with each context element, and the corresponding context-related views are computed. At run time, the context-aware application provides access to the computed portion of interesting information, according to the user’s current active context.

The intrinsic dynamicity of context-awareness constitutes an interesting challenge, and our present efforts are taking into account the issues related to this aspect. In particular, starting from techniques devised in the literature on schema evolution in various fields [16, 3], we are investigating strategies to flexibly support the evolution of the context schema. During the lifetime of the system and application, new context dimensions or values may come into play or become not applicable, as well as constraints may vary. As an example, considering once again the tourism domain, the previously described scenario can be enriched with information about flights, thus the possible topics of interest must be updated. Within this topic, in [23] we propose some evolution operators that keep the coherence between the schema changes and the (induced) changes in the encompassed contexts when necessary.

Furthermore, during the system life also the relevant areas associated with the context elements may need to be changed because different information sources are available or because of fresh context-awareness needs. In the latter case, knowledge discovery techniques may be employed to learn, from the user’s history, the new portion(s) of information that should be replaced as relevant area(s) for (a) given context(s). In studying these problems we have seen that the described modular approach of assigning relevant areas to context elements and allowing the system to combine them automatically facilitates the evolution of the context-aware views in the face of a context-schema change.

The context evolution operators constitute the basis to support context sharing, when more than one peer need to partake of the same context in order, for instance, to set a common understanding of a situation or domain of interest [13]. Other, related challenges concern the automatic synthesis (from observing the environment)
of the possible contexts and thus of the current context schema, or
the automatic recognition of the next active context to the end of
anticipating critical situations [12].

3. CONTEXT-AWARE PERSONALIZATION MANAGEMENT

Data personalization based only on the notion of context repre-
sents a coarse-grained and partial solution to the problem of data
tailing, since the personalization is mostly targeted to classes of
users and does not take into account the individual tastes. For ex-
ample, consider the context of Alice, a young woman currently in
Paris, interested in finding a free hotel room; a contextual system
will suggest hotels close to her current location, with facilities for
young people, and will not be able to propose any ranking or further
filtering of the contextual data according to Alice’s tastes. How-
ever, Alice could be very interested in hotels near the city center
when she is alone, and in Spa hotels (possibly in the city suburbs)
when she travels with friends; thus, a variation in her context, and
in particular in her situation (alone or with friends), may cause Al-
iece’s preferences to change.

In the literature, to improve the personalization task, the knowl-
dege of context is recently coupled with the user personal prefer-
ences: this allows to specify which information represents the ac-
tual interests and needs for a user, in each particular context. In this
direction the use of contextual information has been considered in the
area of recommendation systems [1, 22] to improve the ac-
curacy of recommendations. Moreover, some proposals [26, 18, 15,
17] have addressed the problem of personalizing relational data on
the basis of contextual preferences defined by the user.

However, the majority of these systems require an explicit input
from the user about his/her recommendations or preferences, but
when considering a large variety of data, the manual specification of
an extensive list of preferences for each possible context may be
onerous for the users, who may be discouraged w.r.t. the activity
of explicitly indicating their preferences. This challenge can be
dealt with by automatically inferring the user tastes from his/her past
behaviour, or from the behaviour of users who are recognized as
“similar” to the current one.

In this direction we have used data-mining algorithms to dis-
cover implicit contextual behaviors from analyzing the past data-
querying activity, in the form of association rules that correlate
each context with the values accessed in it. In particular, a σ-rule
ρα on the relation schema R(X) is a triple (C → cond, sup, conf),
where C → cond is an association rule, C is a context and cond
a conjunction of simple conditions in the form A = value, with
A ∈ X, for some Rρα(X) ∈ R(X); sup and conf indicate the sup-
port and the confidence of C → cond, respectively. The support
corresponds to the frequency of C ∧ cond true in the log storing
the user activity; the confidence corresponds to the (conditional)
probability of finding cond true in the log of the activity done in
context C, and is given by sup(C ∧ cond)
sup(C) 2
. For example, if we con-
ider the table HOTEL(NAME, CITY, ADDRESS, SPA, WEBSITE,
...) in the dataset of our scenario, and we mine the σ-rule (user =
woman ∧ situation = with friends → σspa=yesHOTEL,0.8,0.9) by
analyzing Alice’s activity, we can state that, when with friends, she
is often interested in SPA hotels (with a support 0.8 and a con-
idence 0.9).

This approach can be further extended with techniques coming
from recommendation systems, both to prune recommendations not
relevant to the user current context, and to use the extracted context-
tual rules to explain to end users the reason behind the performed
recommendations. Moreover, the proposal can also be applied to
personalize the views for new users (when the history of their ac-
tivity is not available yet) on the basis of the behaviours of similar
users in the same context.

4. DATA-ACCESS MANAGEMENT

In Context-ADDICT the data sources are generally dynamic, trans-
ient, and heterogeneous in both their data models (e.g., rela-
tional, XML, RDF) and schemas 1. Therefore, in order to access
their content, it is necessary to understand their schemas, to repre-
sent them in a suitable semantic formalism (e.g., description logics
or Datalog-like languages [21]) and to map [4] them to some inter-
nal representation of the relevant areas in order to enable context-
aware access.

In Context-ADDICT, the schemas are expressed using the CA-
DL data language that allows to: (i) uniformly represent the data
and the user context(s) in any application domain and (ii) support
efficient context-aware query distribution and answering. Context-
based information filtering is performed at run-time (early tailor-
ing), by selecting fragments of a global representation of the do-
main of interest that are considered relevant for the current context.
These relevant areas are computed by means of the per-context-
element strategy discussed in Section 2 [20].

As noticed in Section 1, in context-aware systems at least some
of the context dimensions require data – the numerical observables
– coming from the applications’ external environment. Therefore,
context awareness has a two-fold role with respect to sensor data:
part of the collected information may contribute to the information
base to be made available through the application and is filtered ac-
cording to context as in the general case; part of it can be directly
used to determine the context itself.

We developed PerLa 2 (PERvasive LAnguage) [24] a declarative
high level language that allows to query a pervasive system, hiding
the difficulties related to the need of handling different technolo-
gies. Originally conceived only for querying sensors, PerLa has
been extended with statements for: i) defining the CDT structure,
also with the capability of acquiring that part of context information
that cannot be deduced from sensors readings; ii) create a context
on a defined CDT; iii) activate/deactivate a context at run-time as
by the actual values of the context variables; iv) perform the con-
textual actions required on the system [25].

However, many problems are still open on this side, both at the
language and at the system level; first of all, the contextual actions
can affect data, procedures and services and, even if, in the current state,
simple ad-hoc operations can be done in PerLa, a general mecha-
nism is still to be built for combining complex actuation sequences
involving whatever kind of action, once a context is active.

At the system level, two major issues are open:

- it might well be the case that the values collected by the sen-
sors are compatible with the simultaneous activation of more
than one context and that these contexts encompass conflict-


1 In our research we mostly concentrate on data sources whose schema is available, while for less structured data sources we re-
sort to techniques found in state-of-the-art literature [14].

2 http://perlaw.sourceforge.net/index.php
5. CONCLUSIONS

We have given a brief report of our understanding of the research on context-awareness, and of the activity on context-aware information systems going on within the PEDiGREE group at Politecnico di Milano. The overall research area is so stimulating that we continuously encounter topics that should be investigated, like the ones mentioned in this paper, which can encourage the scientists who are attracted by this ever-growing research area.

6. REFERENCES


