

# Unified and Conceptual Context Analysis in Ubiquitous Environments

Ali Jaffal, Manuele Kirsch-Pinheiro, Bénédicte Le Grand

Centre de Recherche en Informatique, Université Paris 1 Panthéon – Sorbonne, Paris, France

Ali.Jaffal@malix.univ-paris1.fr, Manuele.kirsch-Pinheiro@univ-paris1.fr, Benedicte.Le-Grand@univ-paris1.fr

**Abstract**— This article presents an original approach for the analysis of context information in ubiquitous environments. Large volumes of heterogeneous data are now collected, such as location, temperature, etc. This “environmental” context may be enriched by data related to users, e.g., their activities or applications. We propose a unified analysis and correlation of all these dimensions of context in order to measure their impact on user activities. Formal Concept Analysis and association rules are used to discover non-trivial relationships between context elements and activities, which, otherwise, could seem independent. Our goal is to make an optimal use of available data in order to understand user behavior and eventually make recommendations. In this paper, we describe our general methodology for context analysis and we illustrate it on an experiment conducted on real data collected by a capture system. Thanks to this methodology, it is possible to identify correlation between context elements and user applications, making possible to recommend such applications for user in similar situations.

**Keywords**—Context Analysis; Recommendation; Formal Concept Analysis; Ubiquitous Computing; Context-aware Systems.

## I. INTRODUCTION

Context management consists in collecting, grouping and exploiting context information on behalf of the user. Several challenges currently affect context management, as for example, the analysis of a large data volume to analyze [1][2]. Indeed, the challenge is no longer collecting data, but to explore it efficiently, which depends on the impact of context on user behaviors and actions. This is particularly important for context-aware systems, whose goal is to adapt their behavior to the user’s context. Several questions arise from this scenario: "How can relevant context information be identified?", "In which context is a specific action performed?", "What is the impact of a given context on user’s actions, and what recommendation can be proposed?".

In this work, we propose a methodology for analyzing the impact of context information on the user actions. We focus in particular on user behavior when using mobile devices, such as tablets or smartphones. Our goal is to provide a way to identify context elements that influence user activities, to understand the relation between them, and to construct a knowledge base connecting actions and context situations.

The proposed methodology is based on Formal Concept Analysis (FCA) [3][4][5] to cluster context and actions based on their relationships. In conjunction with this analysis, we propose to extract association rules in order to recommend particular actions or applications to the user. Association rules are complementary to FCA as they allow quantifying and materializing the causal links between actions and context elements, but also among actions or context elements

themselves. Thus, FCA gathers user actions according to the context under which they have been used in the past, and association rules allow making recommendations of future actions to users in a given context.

This paper is organized as follows: Section 2 discusses related work on context management and describes the basic principles of FCA and association rules. Section 3 describes our original methodology to perform our conceptual analysis of context in ubiquitous systems. In Section 4 we present the results from a real field experiment illustrating the methodology. Finally, Section 5 concludes this paper and presents the perspectives of our work.

## II. RELATED WORK

### A. Context in Ubiquitous Computing

*Context-awareness* stands for the ability of a system to adapt its behavior (operations, services or content) to the current context without explicit user intervention [6][7]. Context-aware systems thus aim at increasing their own usability and effectiveness by taking into account environmental context [6].

Context is a widely concept, as pointed out by Bazire and Brézillon [8] and Coutaz et al. [9]. The most largely accepted definition considers context as *any information that can be used to characterize the situation of an entity (a person, place, or object) that is considered relevant to the interaction between a user and an application* [7]. The relevance of context information is central in this definition and determines its possible use in context-aware systems. According to Greenberg [10], several elements may contribute to the notion of context, and their relevance highly depends on specific situations. Every context-aware system has to determine context elements that will be observed according to its own goals. It is indeed impossible to enumerate in advance a full set of context elements that will apply to any system. This represents an important drawback for these systems, since the relevance of a context element indicates whether this information can be used for adaptation purposes.

Although important, relevance of context information did not receive enough attention on the literature, whose focus is on context management and adaptation. Several context management proposals can be found [2][7][11][12][13], most of them considering context elements and their relevance as predefined a priori during design time. The final purpose of these works is most often adaptation [9][14], namely, adaptation of software components [12], adaptation of supplied content or services [14], adaptation of service composition [13], adaptation by recommending a content or an action according to the user’s context [15]. Whatever the purpose of

this adaptation, it is up to the context management infrastructure to offer all the necessary resources for handling and interpreting context information on context-aware systems.

### B. Context and recommendation

Recommendation systems are personalization mechanisms that can help users find out interesting information or services [15]. Context-based recommendation systems [16][17] try to recommend content or services to users, based on context information observed during previous uses of the system.

Indeed, context information can be seen as a major criterion for recommendation systems [16]. Nevertheless, the notion of context adopted by traditional recommendation systems is often limited. For instance, Pignotti et al. [16] consider as context only time, user's location and history of previously invoked services. Other works, such as [17][18][19], propose recommendation mechanisms that are not limited to particular context elements. Najjar et al. [17] present a prediction mechanism that intends to anticipate user's needs, recommending services according to previously observed context elements, organized in clusters. Similarly, Mayrhofer [18] uses recommendation techniques for anticipating context information and to predict the next likely situation of the user, while Sigg et al. [19] suggest to recommend context information in order to fulfill context description with missing elements based on similar previously observed contexts.

Most context-based recommendation systems use statistical methods and similarity measures for data analysis. Typical data analysis techniques adopted on these works include Bayesian Networks [20] and Markov Chains [17][19] models. Although obtained results are interesting, these methods suffer from some drawbacks. First, classification methods often ignore overlapping classes, preventing context elements to belong to multiple classes simultaneously, even if a context element can be observed in different situations. Besides, classes identified by classification methods are not necessarily understood by final users, which may lead to inappropriate recommendations. Finally, these methods usually require large sets of context data, which are not always available.

In this paper, we focus on context relevance, addressing these issues with a methodology for context analysis based on FCA and on association rules. On the one hand, FCA is a data analysis method that is able to group data at different levels of granularity and to organize them in a coherent set of overlapping classes. On the other hand, association rules allow discovering and quantifying relevant relations among observed values. Although well-known in traditional recommendation systems, FCA and association rules extraction algorithms are not fairly applied to context data. To the best of our knowledge, only a few works [21][22] have tried to apply these approaches to context data. Vanrompay et al. [21] use lattices to group common context data into communities of users, while Ramakrishnan et al. [22] combine Bayesian Networks and association rules to discover frequent correlation between context elements (without recommendation purposes). Indeed, applying these approaches to context data presents some challenges, notably related to data collection and formatting, due to the dynamic and heterogeneous nature of context data. None of these works [21][22] deals with such challenges, contrarily to our methodology. Before presenting it, the next sections introduce underlying analysis methods.

### C. Formal Concept Analysis

FCA [3][4][5] allows performing a conceptual clustering, which helps discovering and structuring knowledge. FCA relies on the lattice theory, which defines a lattice as follows:

**Definition 1:** let  $\leq$  be an **order relation** of a set  $E$ .  $\leq$  defines a total order on  $E$  if all its elements may be compared by  $\leq$ :  $\forall x, y \in E^2, x \neq y \Rightarrow (x \leq y \vee y \leq x)$ . An order which is not total is partial.

**Definition 2:** a **lattice** is a partially ordered set  $(E, \leq)$  where each pair of elements has an upper and a lower bound. A lattice is complete *iff* any part  $S \subseteq E$  has an upper bound (top) and a lower bound (bottom).

From a binary relation between a set of objects and a set of attributes, a **Galois lattice** (or concept lattice) builds a hierarchy of clusters called **formal concepts** [5]. These concepts are built from a table called **formal context**, which expresses the binary relation between objects and attributes.

**Definition 3:** a **binary relation** between sets  $M$  and  $N$  is a set of  $(m, n)$  pairs where  $m \in M$  and  $n \in N$ .  $(m, n) \in R$ , also noted  $mRn$ , means that the element  $m$  is in relation with the element  $n$ .

**Definition 4:** a **formal context** is a triple  $K = (G, M, I)$ , where  $G$  and  $M$  are respectively the set of objects and the set of attributes and  $I \subseteq G \times M$  is a binary relation between  $G$  and  $M$ .  $(o, a) \in I$  means that  $a$  is an attribute of object  $o$ .

Derivation operations  $(.)^I$  are defined for  $O \subseteq G$  and  $A \subseteq M$ :  $O^I = \{a \in M \mid \forall o \in O: o I a\}$  and  $A^I = \{o \in G \mid \forall a \in A: o I a\}$ .  $O^I$  is the set of attributes that are common to all objects of  $O$  and  $A^I$  is the set of objects that have all attributes of  $A$ .

**Definition 5:** a **formal concept** of context  $(G, M, I)$  is a pair  $(O, A)$ , where  $O \subseteq G$  and  $A \subseteq M$ ,  $O = A^I$  and  $A = O^I$ . The set  $O$  is called the **extent** of concept  $(O, A)$  and  $A$  is its **intent**.

**Definition 6:** the set of all formal concepts and the partial order relation between them constitutes a lattice called **Galois lattice** of context  $K$ .

A Galois lattice [5] clusters objects into clusters (i.e., formal concepts) according to their common attributes. A lattice also specifies generalization or specialization relationships among these concepts. Indeed, some of them cluster objects with many common attributes (specific concepts) whereas some contain objects that share very few attributes (generic concepts). The most generic concept (upper bound) contains all objects in its extent, and the most specific one (lower bound) has all attributes in its intent.

In the methodology presented in Section III, a lattice clusters context elements according to user actions and reciprocally. The relationships between context elements and user actions are indeed made explicit in concepts. This allows identifying actions that occur in similar contexts. It also shows correlations among context elements, which is useful in case of missing data. Moreover, in order to quantify causal relations among them, we combine FCA with association rules, described in the following section.

### D. Association Rules

Association rules extraction aims at discovering significant relationships between attributes extracted from databases [23]. Compared to other recommendation techniques, association

rules do not require computing a similarity measure. This is particularly interesting when the context elements and actions are not necessarily comparable, which is our case, since we do not make assumption about context elements.

An association rule is defined as an implication between two itemsets:  $R: X \Rightarrow Y$ , with  $X \subseteq I$ ,  $Y \subseteq I$  and  $X \cap Y = \emptyset$ . The rule  $R$  is said to be based on the frequent itemset  $X \cup Y$  and the itemsets  $X$  and  $Y$  are called, respectively, *premise* and *conclusion* of  $R$ .

To check the validity of an association rule  $R$ , two measures are commonly used:

– *Support*: the support of rule  $R$ , denoted  $\text{support}(R)$  is equal to the frequency of simultaneous occurrences of itemsets  $X$  and  $Y$  i.e.,  $\text{support}(X \cup Y)$ .

– *Confidence*: it expresses the conditional probability that a transaction contains  $Y$ , given that it contains  $X$ . The confidence of rule  $R$ , denoted  $\text{confidence}(R)$  is measured by the ratio  $\text{support}(XUY) / \text{support}(X)$ .

The extraction of association rules consists in determining the set of valid rules, i.e., whose support and confidence are at least equal, respectively, to a minimum support threshold *minsup* and a minimum confidence threshold *minconf* set by the user. This problem is decomposed in [24]: (i) *extraction* of all frequent itemsets with support greater than or equal to *minsup*; and (ii) *generation* of valid association rules (i.e., with confidence greater than or equal to *minconf*) based on the frequent itemsets extracted previously.

We have discussed in this section some related works on context-aware computing, pointing some of its challenges. The relevance and volume of context data that should be analyzed are examples of these challenges, especially for context-based recommendation. Next section presents our methodology that tries to overcome these challenges.

### III. METHODOLOGY

We propose a methodology for unified and conceptual analysis of context, based on FCA and association rules. In this methodology, the impact of context elements is studied in two ways: by clustering context elements with FCA, identifying relationships among them, e.g., to detect redundant data or to complete missing data (due to measurement problems); and by extracting association rules to make explicit and to quantify the strength of relations among context elements themselves and between these and user actions. Our methodology is then divided into three steps: 1) collection and formatting of context elements from the user environment; 2) application of FCA and computation of lattices to structure collected context elements; and 3) extraction of association rules for evaluating the impact of context elements on the user actions, for recommendation purposes. It is worth noting that we focus on discovering user's usual behavior in order to analyze how context elements influence it and then to propose him/her applications (or actions) that he/she is more likely to execute in this context. Different from traditional recommendation systems, we are not interested in influencing the user's choices, but in detecting and reproducing them, similarly to a prediction mechanism.

#### A. Data Collection and Formatting

##### 1) Collection of Context Elements and User Activities:

The starting point of our methodology is a set of raw data collected by sensors or recorded in log files. This step consists in gathering data related to user activities (i.e., applications executed on a mobile device) and environment (e.g., temporal information, location, network connection, etc.).

Storing contextual data is necessary for computation of lattice and association rules, as those are based on previously observed data. This data collection should, of course, respect privacy legislation, in particular in terms of explicit user agreement and anonymization. No assumption is made about the context elements we collect. Our approach considers, in a unified way, any context element. Potential interdependencies (or redundancies) among them will be identified during the analysis described in the following sections.

##### 2) Data Formatting:

Collected raw data must be formatted in order to be processed by FCA. The input data is organized as a set of user activities (our objects) and a set of Boolean attributes, corresponding to observed values of context elements. During this phase, temporal data (i.e., timestamps) are transformed into time intervals, and location information (e.g., GPS coordinates) into geographical zones. At the end of this pre-treatment phase, each user activity can be associated to obtained values. Next step aims at extracting implicit relationships among contextual and activity data.

#### B. Extraction of Relationships among Data with FCA

As explained in Section II, FCA is a mathematical method that clusters data into concepts in lattices. We use FCA to organize contextual information into overlapping classes at different levels of granularity. Unlike other analysis methods, FCA can find a natural data structure, combining the user actions to the context elements observed during previous uses. This structure enables the connection of contextual data with user actions and allows building a knowledge base (e.g., actions 1 and 2 are always executed in a similar context, which may suggest some proximity between these actions). Besides, the obtained concept lattice translates the hierarchical relationships between formal concepts, and can be used for classification and prediction purposes.

##### 1) Formal Context Specification:

This step consists in identifying the data elements that will become the objects and attributes in a formal context. Formal context corresponds to a table, similar to Table I, that combines the activities performed by a user (objects  $A_i$ ) and the corresponding context elements (attributes  $C_j$ ). Thus,  $(A_i, C_j)$  indicates if the activity  $A_i$  has been performed in the presence of the context element  $C_j$ . For instance, activity  $A_2$ , in Table I, has been performed in contexts  $C_2$  and  $C_3$ .

TABLE I. MATRIX REPRESENTING A FORMAL CONTEXT

Attributes Objects	C1	C2	C3
A1	1	0	1
A2	0	1	1
A3	1	0	0

## 2) Construction of Galois Lattices and sub-lattice:

The formal context specified above is used to build a Galois lattice, clustering user activities and observed context elements, as illustrated in Figure 2.

When the lattice is not too big, its graphical representation can be interpreted visually to identify relationships between objects and attributes. However, the lattice grows fast when the number of objects and attributes increases. It is thus necessary to divide it into sub-lattices, by dividing context elements into subsets and computing the corresponding sub-lattices.

The cross-interpretation of sub-lattices allows identifying semantic links between context elements. Each sub-lattice (focused on a given context element) brings knowledge about activities conducted in this context and about correlations with other context elements. This information is extremely useful to complete missing data due to problems during data capture.

However, the analysis based on FCA has some limitations. First, the choice made when splitting context elements into sub-lattices may hide some relationships, which become more difficult to see depending on the way attributes have been separated. The second limit is that causal links between context elements and user activities are not quantified in lattices concepts. In other words, when multiple context elements are observed for a given activity, the lattice does not tell whether a context element is more “important” than another. The extraction of association rules, conducted in the following step, answers this question and can be used to propose recommendations.

### C. Association Rules Extraction for Recommendation

Since activities are often used together with other activities, those can be considered as context elements as well. Indeed, there is often a semantic link between consecutive activities. This link is particularly interesting for recommendation purposes and thus for the extraction of association rules. We have, therefore, considered for this work that any activity conducted within an interval of 30 minutes before a given instant belongs to the user context at this instant. These results form an enriched formal context, which is used as input data for the identification of association rules.

Before extracting association rules, we define two variables used to filter input data. Indeed, activities and context elements that are too frequent may hide interesting association rules among less frequent ones.

*Activity frequency* =  $n^\circ$  of observations in which an activity appears / total  $n^\circ$  of distinct values of (extended) context elements

*Context element frequency* =  $n^\circ$  of observations in which this context element is associated to an activity / total  $n^\circ$  of distinct observed activities

We have used the well-known Apriori [24] algorithm for the construction of frequent itemsets and the extraction of association rules. This algorithm operates in two phases: it first identifies the frequent itemsets that have a minimal support, then analyzes them to determine association rules whose confidence index is superior to a given threshold.

We consider that association rules with sufficient confidence value can be used for recommendation purposes. A recommendation such as  $C_1, C_2, C_3, A_4, A_5 \Rightarrow A_6$  means that if a user is in the context  $(C_1, C_2, C_3)$  and has recently made actions

$A_4$  and  $A_5$ , then action  $A_6$  is recommended to him/her (as it is very likely that he/she will perform it). The recommendations based on association rules allow anticipating the next action of a given user based on previous usage.

We have applied our methodology to a case study, presented in the following section.

## IV. CASE STUDY

In order to demonstrate our methodology, we applied it in a real data set, collected by a capture system on a user’s tablet. The capture system is an Android application running in the background, which observes at regular time intervals the applications used and their execution context, without interfering with them. We have experimented this capture system with a single user during 69 days. The collected data has been stored into a SQLite database. For this first experiment, we have decided to consider just one user in order to better evaluate with him the results obtained with FCA and association rules.

### A. Data Collection and Formatting

#### 1) Collection of Context Elements and User Activities:

The capture system collects information about: (i) *applications* launched by the user on her/his device (e.g., Facebook, Maps, Dropbox, etc.). We have identified 47 distinct applications of different categories (news, games, social networks, etc.); (ii) *geographical locations* visited by the user, which correspond to observed GPS coordinates; (iii) information on internal and external *memory* states of the device; (iv) *networks* to which the user has been connected over time. The capture system periodically observes these context elements, associating each observation to a *timestamp*. According to privacy legislation, the user involved in this experiment has been informed of the data collection process and has a full access to collected data, since these are locally stored in his personal device. The user also keeps full control of the collecting application, actively launching it.

#### 2) Data Formatting

From the raw data collected in the previous phase, we have created the objects (corresponding to applications) and attributes (corresponding to observed context elements) needed to define formal contexts. These measured values cannot be taken into account as they are, since lattices require Boolean attributes. This is not a problem for network data, as for example the *Network\_1* attribute is set to 1 if the user is connected to that network and to 0 otherwise. The possible values of (internal and external) memories are 1GB, 2GB and 3GB. We create, therefore, 6 attributes (*MemInt1*, *MemInt2*, *MemInt3*, *MemExt1*, *MemExt2*, *MemExt3*).

In the following, we detail the processing of temporal and geographical data, which are more complex.

#### 3) Temporal data processing

We have transformed the measured timestamps into 6 time intervals per day: morning ( $t_{Morning}$ : 6h-12h), noon ( $t_{noon}$ : 12h-13h), afternoon ( $t_{Afternoon}$ : 13h-20h), evening ( $t_{Evening}$ : 20h-00h), night ( $t_{Night}$ : 00h-06h). These intervals can, of course, be different depending on users and on their behavior. Each time range corresponds to a Boolean attribute, associated to applications used on the device.

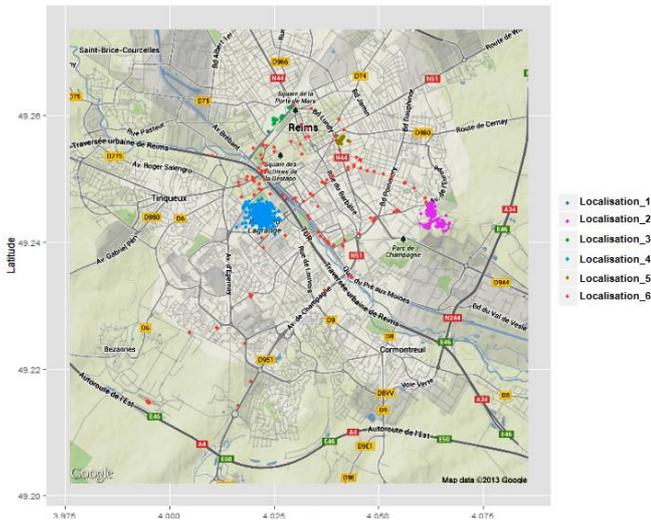


Figure 1. Zoom on the use of the tablet in a city.

#### 4) Geographical data processing

We have processed collected geographical data in order to map them to relevant zones. We have identified these zones using R software, instead of dividing the longitude and latitude data into regular rectangular zones that would not be meaningful. The strength of this approach is that the number of zones is not fixed in advance and these zones do not need to have the same surface. Figure 1 shows the locations observed in our experiment. We may notice that most locations belong to a zone, whereas others places are visited much less frequently by the user. The mapping between each point and applications used at this location is then achieved later according to the identified geographical zones.

We identify all points in the dataset, which belong to a dense zone. Points that are not associated to a dense zone are then studied, in order to see if new dense zones appear. At the end of this process, points that are not associated to any zone are considered as movement locations (on the path between two zones), corresponding to a new zone. We have finally labeled each zone (*Location\_1*, *Location\_2*, etc.).

#### B. Formal Concept Analysis

Several tools exist for building Galois lattices, such as *Lattice Miner* [25] and *Conexp* [26]. We have used *Conexp* with the formal context described in previous section and built the associated lattice, illustrated in Figure 2.

As explained in Section III, the direct interpretation of the whole lattice may be difficult. We have therefore divided the original formal context and built sub-lattices. We have built the 3 sub-lattices corresponding to attributes related to location, networks and time periods respectively. Figure 3 shows the sub-lattice built from location attributes. It contains concepts corresponding to groups of applications used in various locations. This sub-lattice is much more readable than the whole lattice of Figure 2 and can be interpreted visually.

In order to know the applications that are used in a given geographical zone (e.g., *Location\_2*), we only have to find the corresponding concept in the formal context and identify the applications in its extent, as well as all inheriting applications (below that node). For instance, the applications used at the

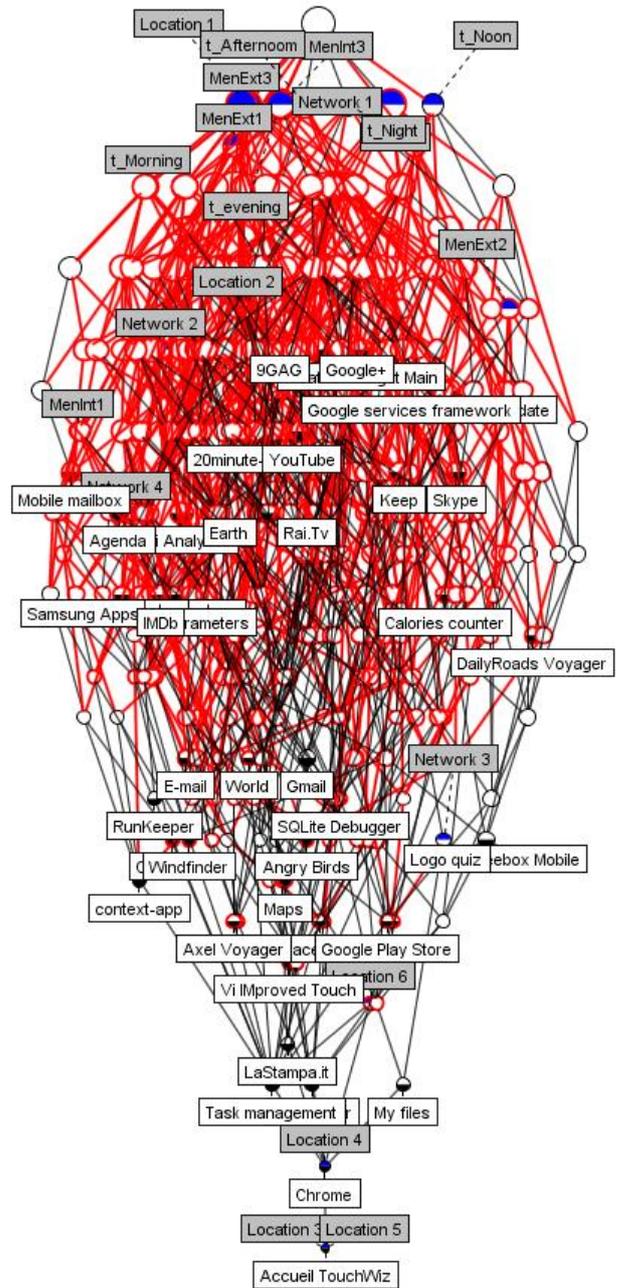


Figure 2. Global concept lattice (with all context elements).

*Location\_2* are *ConnectBot*, *camera*, *Drive*, *E-mail*, *Calendar*, *Chrome*, *TouchWiz*, etc. Applications that are common to two locations appear in a new node, which inherits from original nodes. The applications that have been used both at *Location\_2* and at *Location\_3* are *Calendar*, *Chrome* and *TouchWiz*. The lower a concept is in the lattice, the more specific it is, i.e., it contains more attributes in its intent. We proceed similarly to build the sub-lattice related to connection networks (Figure 4).

It should be noted that dysfunctions of the capture system may result in missing information. Some applications could thus not be associated to any context attribute. Therefore, they only appear in the upper bound of the lattice, which contains no attribute in its intent (cf. Figure 3 and Figure 4). For example, no location could be associated to the applications: *Alarm*,

Google+, Skype, Youtube... Likewise, no access network has been captured for the calculator application.

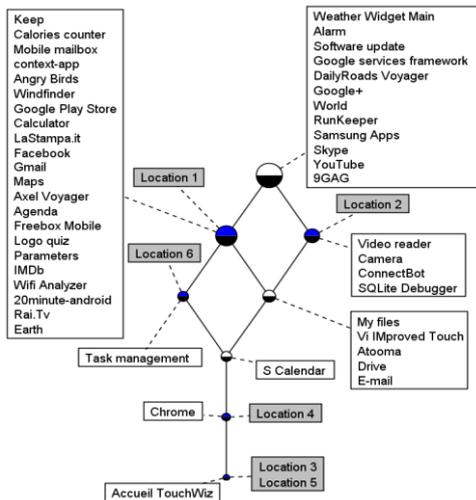


Figure 3. Sub-lattice corresponding to the locations context.

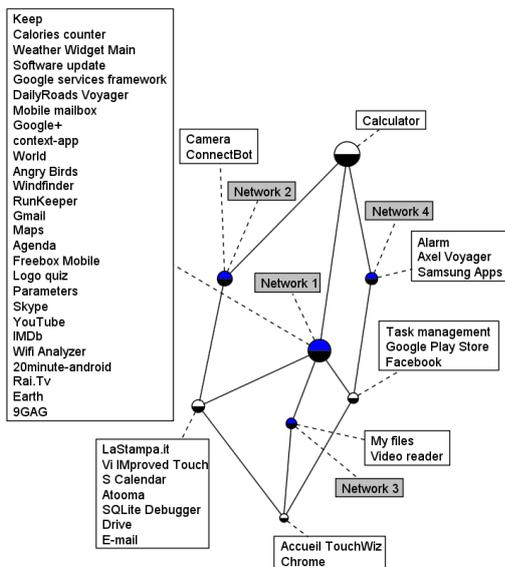


Figure 4. Sub-lattice corresponding to the network context.

We notice a strong relationship between *Location\_1* and *Network\_1* contexts, through the numerous intersections between these context attributes in terms of related applications (*Maps, Gmail, Agenda, etc.*), as shown on Figures 3 and 4. This relationship has been validated by the user, who confirmed that indeed *Network\_1* is physically located in *Location\_1*. Intersections like this one allow recovering missing information, i.e., information that could not be captured. We can, for instance, infer the access networks associated to the *calculator* application (on the top of the lattice in Figure 4), since it is often used in *Location\_1*, which can be associated with the *Network\_1* context.

C. Association Rules Extraction for Recommendation

During this step we have used an extended formal context as an input, in which we have considered recent applications as part of the user context, in addition to temporal, geographical and network connection attributes. We have also discarded very frequent applications, which do not bring relevant information, such as system applications. The frequency diagram of remaining applications is presented in Figure 5; applications with high frequencies are used in a significant proportion of contexts. This is the case for *Chrome*, with a frequency equal to 0.65.

Figure 6 shows the frequencies of context elements. When the frequency is high, the corresponding context element is frequently associated to applications. For example, *Location\_1* (which is the user’s home) has a frequency of 0.65, which means that many applications are used from there.

We have applied the Apriori algorithm [24] to the extended and filtered applications and context elements. Apriori first computes the set of frequent itemsets together with their support measure, such as:

$$E1 = [E-mail, Gmail, Rai.Tv, 20minute-android, t_Evening, Network_1, Google+], supp:11.62\%$$

*E1* is a frequent itemset (both context elements and applications) with a support equal to 11.62 %. *E1* is a set of context elements for the user.

We have obtained about a hundred frequent itemsets, such as the ones shown in Table II. We have only kept the itemsets with a support superior to 10% for the extraction of association rules. The choice of a low support value allows considering a large fraction of frequent itemsets (further filtering is made later, as explained in the following). Among all generated rules, user has rejected only a small set (about 23%), most of

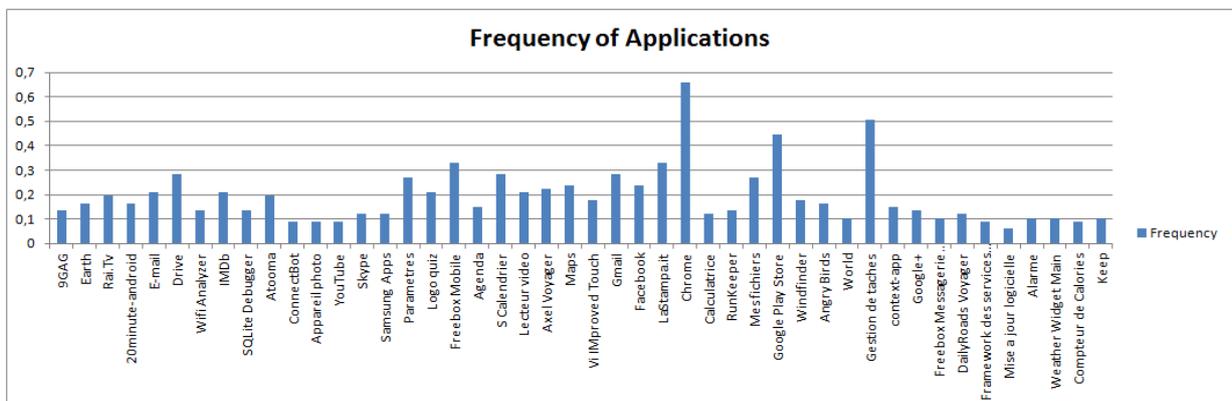


Figure 5. Applications frequency diagram

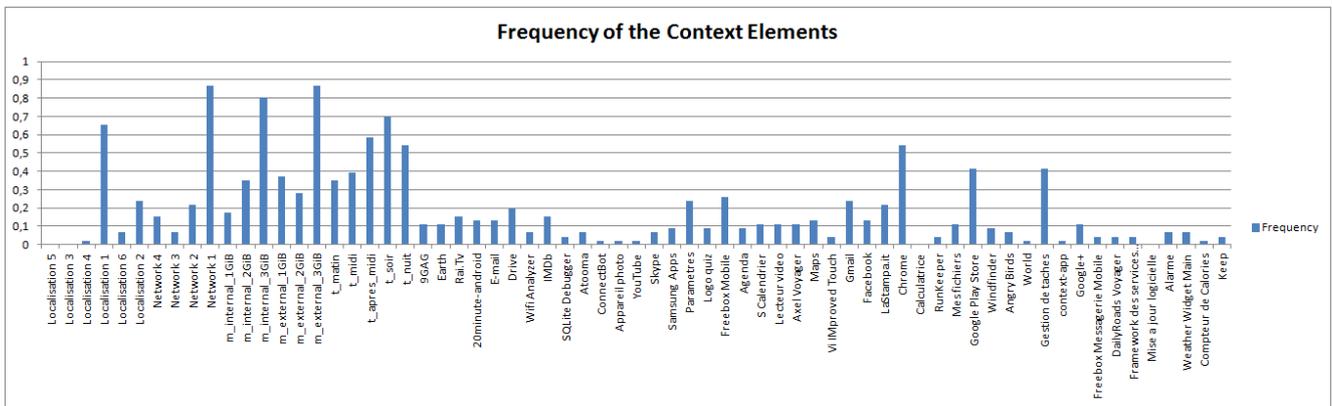


Figure 6. Context elements frequency diagram.

them with lower confidence values. When considering confidence above 75%, rejection decreases to 18,5%, which represents a promising result for us.

TABLE II. EXAMPLE OF FREQUENT ITEMSETS.

FI id	Frequent itemset	Supp
EIF69	[t_Evening, Chrome, Facebook]	13.95%
EIF95	[t_Evening, Network1, Maps, 9GAG]	11.62%
EIF100	[Network1, LaStampa.it]	25.58%
EIF101	[t_Evening, Localisation1, Network1, LaStampa.it]	20.93%
EIF106	[t_Evening, Network1, Gmail]	23.25%
EIF112	[t_Evening, Network1, Chrome]	34.88%
EIF117	[t_Night, Chrome]	37.20%
EIF118	[Network1, Chrome]	46.51%

TABLE III. EXAMPLES OF RECOMMENDATIONS

	Recommendations (association rules)	Conf
R1	[E-mail, Gmail, t_Evening, Network1] $\Rightarrow$ [Google+]	83.33%
R43	[Network1] $\Rightarrow$ [Google+]	13.51%
R57	[E-mail, t_Evening, Network1] $\Rightarrow$ [Google+]	83.33%
R71	[t_Night, Localisation1, Chrome] $\Rightarrow$ [Facebook]	35.71%
R82	[Network4] $\Rightarrow$ [Facebook]	83.33%
R86	[t_Evening, Network4, Chrome] $\Rightarrow$ [Samsung Apps]	83.33%
R95	[t_Evening, Network1, Maps] $\Rightarrow$ [9GAG]	100.0%
R101	[t_Evening, Localisation1, Network1] $\Rightarrow$ [LaStampa.it]	45.0%
R109	[t_Evening, Localisation1] $\Rightarrow$ [Gmail]	40.90%
R123	[t_Afternoon, Localisation1] $\Rightarrow$ [Chrome]	61.11%

From these itemsets, we have extracted all association rules. Then, we have eliminated (filtered) all rules whose conclusion is a context element, as our goal is to recommend applications. We have also filtered all the rules that contain incompatible contexts in their premise, e.g., with two different locations at the same time, e.g.,  $[t\_Afternoon, t\_Evening, Network1] \Rightarrow [Chrome]$ . This rule will never be used as the user will never be simultaneously in the afternoon and in the

evening. Table III shows a sample of recommendations for our case study (with different confidence values).

We have obtained in total 144 recommendations, 38 of which rely on association rules with a confidence equal to 1: these recommendations correspond indeed to the behavior identified by the user himself. Moreover, 103 recommendations have a confidence greater or equal to 50%.

## V. CONCLUSION AND FUTURE WORK

In the approach presented in this paper, we have used FCA for the management of context and association rules for making recommendations. We have described existing context management approaches and shown their limitations. We have presented our methodology for context management, based on the analysis of formal contexts, the construction of Galois lattices and the extraction of association rules, in order to study the relationships between user actions and contextual information and to be able to give recommendations to users. We have described FCA in mathematical terms for explaining our method and the different underlying notions. Based on this theory, we have proposed a methodology for context analysis consisting of 3 steps. A data filtering and formatting are performed first in order to extract a formal context and thereafter build a lattice. The itemsets of this lattice are then interpreted with association rules to make appropriate recommendations and facilitate decision making. However, if the global lattice is too large, a decomposition into sub-lattices allows performing a visual analysis and making both an individual interpretation of each sub-lattice and a cross interpretation.

We have applied our methodology to a case study based on real data: we have used the data obtained by a capture system installed on the tablet of a user. The results have provided important information on the context of application usage, as well as relations between the different context elements. We have thus deduced information about the applications on all dimensions (contexts). We could also make appropriate recommendations with association rules. We could also complete the missing data due to occasional dysfunctions of the capture system.

The approach and solution proposed in this article open many perspectives for future work. The first one consists in devising mechanisms to automate cross-interpretations and associated recommendations. We will apply them to the concepts generated by the Galois lattice and all links between

these concepts, so as to automatically deduce an interpretation and recommendations with association rules.

We have used so far a limited number of context elements (geographical location, time, network connection and device memory). In the future, we will study the relevance of the other types of context elements to extend our approach. The same applies to users. Indeed, we are currently extending our case study to several users with different profiles (age, professional activity, etc.). We will also extend our experiment to include both automatic and non-automatic data collection, in order to identify other context elements that could be observed.

As future work, we will try to build the relations between actions and contexts themselves, and the relations between users' profiles. Thereby we seek to model a user profile (age, sex, student/employee, needs), according to available information, used applications, and moments, and then add information about the applications according to the different categories (Games, News, Entertainment, Economics, Education, Finance, Books, Weather, Sports, Travel), then define recommendations (Marketing, Increase sales, increase users' satisfaction, increase the audience). With this additional information we can provide new predictions and recommend new actions.

We will also apply our methodology to other case studies with more users, and we will evaluate the recommendations from them. In the longer term, we wish to manage context in real time, which will raise scalability issues due to the volume of data to be analyzed and to temporal constraints. A possible track consists in using distributed approaches for the construction and the update of the Galois lattice.

#### REFERENCES

- [1] C. Bettini et al., "A survey of context modelling and reasoning techniques". *Pervasive and Mobile Computing*, vol. 6, no. 2, Elsevier Science Publishers, April 2010, pp. 161–180.
- [2] J.P. Arcangeli et al., "INCOME - Multi-scale Context Management for the Internet of Things", *AmI '12 : International Joint Conference on Ambient Intelligence*, Pisa : Italy, 2012, pp. 338-347.
- [3] R. Wille, "Formal Concept Analysis as Mathematical Theory of Concepts and Concept Hierarchies". *Formal Concept Analysis*, B.Ganter et al., eds., pp. 1-33, Springer-Verlag, 2005.
- [4] U. Priss, "Formal Concept Analysis in Information Science". In: Blaise, C. (ed.) *Annual Review of Information Science and Technology*, ASIST, vol. 40, 2006, pp. 521-543.
- [5] A. Guénoche and V. Mechelen, "Galois approach to the induction of concepts". In I. Van Mechelen, J. Hampton, R.S. Michalski, & P. Theuns (Eds.), *Categories and concepts: Theoretical and inductive data analysis*, London: Academic Press, pp. 287-308, 1993.
- [6] M. Baldauf, S. Dustdar, and F. Rosenberg, "A survey on context-aware systems". *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 2, no.4, 2007, pp. 263-277.
- [7] A.K. Dey, "Understanding and Using Context". *Personal and Ubiquitous Computing*, vol. 5, no. 1, 2001, pp. 4-7.
- [8] M. Bazire and P. Brézillon, "Understanding Context Before Using It". In: Dey, A.K., Kokinov, B., Leake, D.B., Turner, R. (eds.) *CONTEXT 2005*. LNCS (LNAI), vol. 3554, 2005, pp. 29–40.
- [9] J. Coutaz, J. L. Crowley, S. Dobson, and D. Garlan, "Context is key". *Communications of the ACM*, vol. 48, no. 3, 2005, pp. 49-53.
- [10] S. Greenberg, "Context as a dynamic construct". *Human-Computing Interaction*, vol. 16, no. 2-4, 2001, pp. 257-268.
- [11] D. Conan, R. Rouvoy, and L. Seinturier, "Scalable Processing of Context Information with COSMOS", *Proc. 7th IFIP International Conference on Distributed Applications and Interoperable Systems*, Springer-Verlag, Lecture Notes in Computer Science Volume 4531, Paphos, Cyprus, June 2007, pp. 210-224.
- [12] M.Wagner, R. Reichle, and K.Geihls, "Context as a service - Requirements, design and middleware support". *Int. Conf. on Pervasive Computing and Communications Workshops*, 2011, pp. 220-225
- [13] N. Taylor et al., "Pervasive Computing in Daidalos", *Pervasive Computing*, vol. 10, no.1, 2011, pp. 74 -81.
- [14] J. Gensel, M. Villanova-Oliver, and M. Kirsch-Pinheiro, "Modèles de contexte pour l'adaptation à l'utilisateur dans des Systèmes d'Information Web collaboratifs" ("Context models for user adaptation on collaborative Web Information Systems"). 8<sup>èmes</sup> J. Francophones d'Extraction et Gestion des Connaissances (EGC'08), *Atelier sur la Modélisation Utilisateur et Personnalisation d'Interfaces Web (in French)*, 2008, pp. 5-15.
- [15] Q. Wen and J. He, "Personalized Recommendation Services Based on Service-Oriented Architecture". In: *IEEE Asia-Pacific Conference on Service Computing*, 2006, pp. 356–361.
- [16] E. Pignotti, P. Edwards, and G.A. Grimnes, "Context-Aware Personalised Service Delivery". *European Conference on Artificial Intelligence*, ECAI, 2004, pp. 1077-1078.
- [17] S. Najar., M. Kirsch-Pinheiro, Y. Vanrompay, L.A. Steffeneel, and C. Souveyet, "Intention Prediction Mechanism In An Intentional Pervasive Information System". *Intelligent Technologies and Techniques for Pervasive Computing*, 2013, pp. 251-275.
- [18] R. Mayrhofer, "An Architecture for Context Prediction". PhD thesis, Johannes Kepler University, 2004.
- [19] S. Sigg, S. Haseloff, and K. David, "An Alignment Approach for Context Prediction Tasks in UbiComp Environments". *Pervasive Computing*, vol.9, no. 4, 2010, pp. 90-97.
- [20] A.K. Ramakrishnan, D. Preuveneers, and Y. Berbers, "A Loosely Coupled and Distributed Bayesian Framework for Multi-context Recognition in Dynamic Ubiquitous Environments", *10th International Conference on Ubiquitous Intelligence & Computing and 10th International Conference on Autonomic & Trusted Computing (UIC/ATC 2013)*, IEEE, 2013, pp. 270-277.
- [21] Y. Vanrompay, M. Kirsch Pinheiro, N. Ben Mustapha, and M.-A. Aufaure, "Context-Based Grouping and Recommendation in MANETs", In K. Kolomvatsos, C. Anagnostopoulos, and S. Hadjiefthymiades (Eds.), *Intelligent Technologies and Techniques for Pervasive Computing*, IGI Global, pp. 157-178, 2013.
- [22] A.K. Ramakrishnan, D. Preuveneers, and Y. Berbers, "Enabling Self-learning in Dynamic and Open IoT Environments", *The 5th International Conference on Ambient Systems, Networks and Technologies (ANT-2014)*, the 4th International Conference on Sustainable Energy Information Technology (SEIT-2014), *Procedia Computer Science*, vol. 32, 2014, pp. 207-214.
- [23] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases". In *Proceedings of the ACM SIGMOD Intl. Conference on Management of data*, Washington, USA. June 1993, pp. 207–216.
- [24] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules". *Proceedings of the 20th International Conference on Very Large Databases*, June 1994, pp. 478–499.
- [25] B. Lahcen and L. Kwuida, "Lattice Miner: A Tool for Concept Lattice Construction and Exploration". In: *8th Int. Conf. on Formal Concept Analysis*, Agadir, Morocco <http://lattice-miner.sourceforge.net/> [retrieved: June 2014], 2010.
- [26] S.A. Yevtushenko, "System of data analysis "Concept Explorer"". In: *Proceedings of the 7th national conference on Artificial Intelligence*, Russia, <http://conexp.sourceforge.net/> [retrieved: June 2014], pp. 127–134 (in Russian), 2000.