Evaluation and Selection of Context Information

Maria Chantzara, Miltiades Anagnostou

Computer Networks Laboratory, School of Electrical & Computer Engineering,
National Technical University of Athens (NTUA),
9 Heroon Polytechniou Str, 157 73 Zografou, Athens, GREECE
{marhantz,miltos}@telecom.ntua.gr

Abstract. The easy creation of context-aware services requires the support of management facilities that provide ways to more easily acquire, represent and distribute context information. The contribution of this paper refers to the challenge of evaluating and selecting the context information that is used by the services. It proposes an approach that trades off economic cost, user preferences and offered context information. According to this, there is no need to know beforehand the context providers to obtain the required information, but a more ad-hoc discovery and organization of the providers is envisioned.

1 Introduction

The advances in wireless communications and user mobility have given quite a boost to the research about new classes of applications, namely the Context-Aware Services (CASs) that get aware of the execution environment such as location, time, user's activities, devices' capabilities in order to tune their intended functionalities and adapt to the changing environment and user requirements. The development and provisioning of CASs need to be supported by management facilities capable of collecting, manipulating, reasoning and disseminating context information. In this respect, researchers have been building tools and architectures to tackle these issues and facilitate the easy creation of CASs. The contribution of this paper refers to the challenge of evaluating and selecting the context information to be used by the CASs.

As research on context-awareness evolves, new context sources are expected to come and go rapidly, and context requests will increase. To make matters worse, among the available context information sources, there are sets of sources that provide information referring to the same entity, but are produced based on different sensing technology or/and processing technique under the administration of different business entities. As a result, even though the information may refer to the same entity, it varies in its update frequency, its accuracy, its format of representation, and its price. Moreover, the sensed context information can be inaccurate or unavailable due to sensor or connectivity failures, while user-supplied information is subject to human error and staleness. The reliance of CASs to imperfect information can often cause usability problems [0]. On the other hand, CASs are characterized by a higher level of mobility since CAS users tend to move among different heterogeneous environments where different context providers operate and provide information.

Furthermore, the provisioning of the same context information with different characteristics calls for advanced personalization features that can lead to the provisioning of sophisticated CASs. Specifically, during subscription the user enters the parameters that personalize the service operation. These parameters form the user's service profile that encompasses his/her needs from the CAS. For example, a user that subscribes to a context-aware restaurant finder service could specify that he/she wishes the weather conditions to be taken into account in contrast to another user that just wants to find what are the restaurants in the city that match his/her taste preferences. In addition to the profile, the user identifies the price ceiling for the personalized service provisioning and awaits best possible service delivery within his/her price range [2]. From service management point of view this connotes a complete inversion: not the service defines the price but the price causes the type of service provisioning. The price ceiling along with the subscription parameters determines the type of service to be offered and the information to be obtained.

As a consequence of the aforementioned issues, a new challenge referring to the evaluation and dynamic selection of the context information to be utilized by the CASs arises. Even though many research efforts [3] have been conducted to specify and materialize frameworks and toolkits for context-awareness, these efforts lack to deal with the provisioning of CASs in multiple environments that reveals the necessity to elaborate on issues that have to do with the collection of data from multiple context sources maintained by different administrative entities [4]. In addition, these efforts either assume that CASs rely on the context information that is accurate and up-to-date [5], not taking into consideration the quality of context information at all, or just deal with data quality as properties of the information to be collected [6]. Additionally, many researchers try to deal with imperfect context utilizing various techniques mostly coming from artificial intelligence consuming resources. A careful assessment trading-off their success in removing context ambiguity and their cost should be made in order to determine whether these techniques worth be used on behalf a specific CAS.

In this paper, we explore a novel approach for the evaluation and selection of context providers that tries to balance quality of information, economic cost and user preferences. This approach requires the enhancement of context management frameworks with the *Context Matching Engine* (CME) that performs the proposed multicriteria decision making. Therefore, there is no need to know beforehand the providers to obtain the required information, but a more ad-hoc discovery and organization of the context providers is envisioned. It also enables flexibility to failures of context providers or appearance of new ones. In the rest of the paper, we present the proposed approach. The paper is organized as follows: In Section 2 the functionality of Context Matching Engine is analyzed. In Section 3, the problem of selecting the context information to be utilized by a CAS is modeled and formally described. The problem solution and some initial results are presented in section 4. Finally, Section 5 provides some conclusive remarks and future plans.

2 Context Matching Engine

Context information goes through four lifecycle phases till a CAS can access it. The first phase is context sensing that represents the acquisition of context from the

Context Information Sources e.g. sensors or network resources. Until the CASs can actually use the data, context processing and dissemination take place. Context processing includes the generation of high-level context information from primitive data by enforcing functions of interpretation, filtering, aggregation and inference. Context dissemination refers to the efficient distribution of context to the CASs. The context lifecycle ends with the context usage that describes the utilization of context by CASs in order to trigger the appropriate actions. The aforementioned phases introduce new opportunities in the market arising out of the exploitation of context information trade. The different business roles participating in CAS provision are depicted in Fig.1.

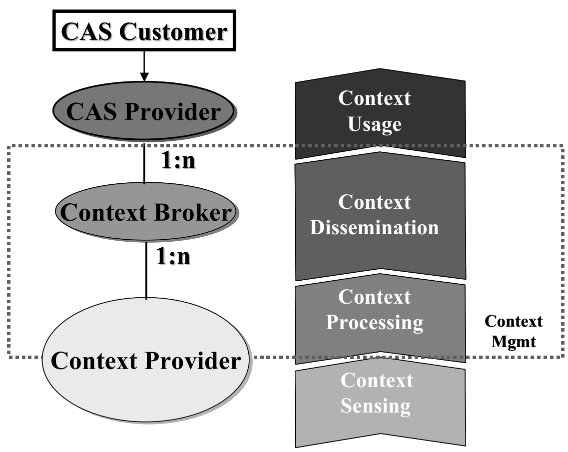

Fig. 1. Context phases and involved actors

A CAS Provider is responsible for providing CASs to the customers/users and managing them. The Context Provider deploys and operates the various Context Information Services that communicate with the sources and sense or process information. Finally, the role of Context Broker is responsible for handling the distribution of context data to the CAS Providers or directly to the CASs. This role provides efficiency and reduces time-to-market for CASs. CASs could interact with the Context Providers to acquire the necessary context information, but this would acquire more programming effort for the CAS developers. We have implemented such architecture accommodating the role of Context Broker as part of our work in the CONTEXT project [7]. According to this, the Context Broker offers the *Context Information Provider Interface* that enables context providers to declare and supply the context information they expect to provide and the *Context Information User Interface* that enables CASs to request the information they want to consume. Context Brokers act as a federation that cooperates to answer context requests. A common model of context information is required so that producers and consumers of information have the same understanding. For this reason, CONTEXT has introduced the notion of Context Modules, which are XML-based descriptions containing the properties of the context information supplied by the Context Providers. As in data-centric approaches, it is not required CASs to name a specific Context Provider from which they wish to retrieve some information but just the information they wish to obtain. Finally, the Context Broker is responsible for discovering and then interacting with the appropriate information producer at runtime.

As it is already mentioned in previous section, the richness of context information gathered from sensors and human users as well as the need to execute CASs in multiple environments require decision-making regarding the exact information that a CAS should use. In order to tackle these issues, we introduce the Context Matching Engine (CME) operating in each domain under the administration of Context Broker. The selection balances the fulfillment of the service objectives defined by the users and the services, and the constraints imposed by the given price ceiling. In order to perform the selection of context information, the CME accommodates the process of *Context Validation* that monitors the validity of the offered context information and concludes about the fidelity and the time response of the context producer as well as the *Context Evaluation* process that considers the selected service profile, the user subscription data, and the outcomes of the context validation process and computes the expected utility of each offered context information. The results of the aforementioned processes along with the imposed constraints facilitate as input data to a *decision-making algorithm* that determines the appropriate information to be utilized by the context-aware service. It should be mentioned that after the selection is made, the CME monitors the status of the offered context information and decides the switch of context providers on behalf of the CAS when new context providers are registered or one of the already selected fails or one of the non selected is upgraded. The logical decomposition of the CME is depicted in Fig. 2.

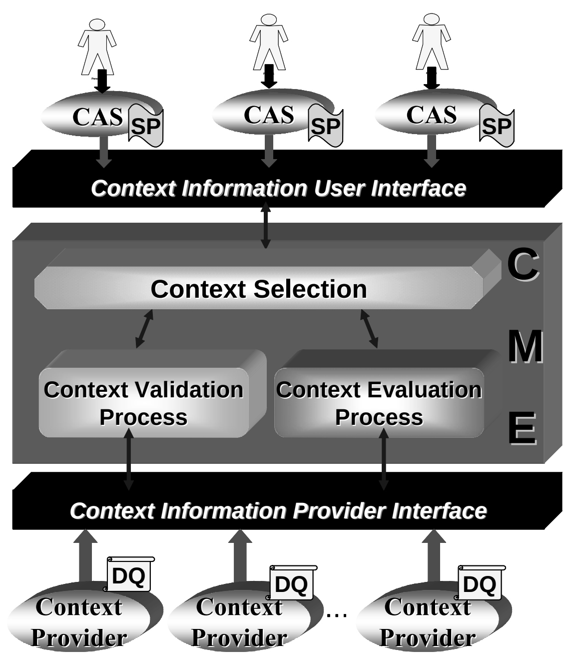

Fig. 2. Context Information Selection

Context providers utilize the “Context Information Provider Interface” offered by the Context Broker to register the type of information they provide accompanied with the data quality attributes (DQ) that characterize it. The different nature of the context information complicates the identification of general quality parameters [8]. Considering work in [9],[10], we conclude to the following attributes that characterize context information and may affect context-aware service behavior: (a) Accuracy: It describes how the information describes the reality. It depends on the resolution pattern that is used and the smallest perceivable element. (b) Precision: It describes the degree of confidence that the given value of context and the corresponding accuracy is given. It shows how often context information is unintentionally wrong because of internal problems. (c) Time refresh: It describes the time period that a new measurement of the specific data attribute is done. It is of special interest for the case that the information is highly dynamic in time. (d) Time sample: This attribute describes the exact time point that the value has been obtained. It is obvious that the complete list of attributes does not refer to all types of context information. Specifically, it mostly characterizes dynamic information that changes frequently over time and is produced by physical or logical sensors. On the other hand, the quality of static information such as user’s profiles and preferences is determined by the time sample. Additionally, it is essential to determine some quality parameters that characterize the producers of the information and vary with time, network topology and characteristics. These are: (a) Fidelity: It depicts the ambiguity of the registration parameters that the context provider specifies. In specific, it is a probability of correctness that is calculated by the Context Validation Process based on the previous behavior of the context provider. Accordingly, those producers that have not been detected as incorrect tend to be more trusted by the consumers. Therefore, statistics of context provider’s behavior need to be gathered in order to determine its fidelity. (b) Time response: It corresponds to the time interval that is required by the producer of information in order to reply to the application’s request. It is particularly useful for time criticality issues. Apart from the technical details of the context provider, this parameter depends to the topological properties of context consumer and provider and the intermediate Context Brokers.

CASs utilize the “Context Information User Interface” and declare the context information to be retrieved as specified by the service profile (SP). Along with the context information to be collected, the services indicate the constraints regarding the data quality of the context information, as well the time reaction of the service and the economic cost. In addition to these, the CAS also specifies the service policies that describe how CME should behave in case user requirements can not be satisfied by the available context sources. In response to the context request, the CME triggers the Context Evaluation Process that evaluates the available context sources and finally the optimal sources to provide the requested data is determined. Based on this decision the appropriate source is contacted and the required data are retrieved.The context sources deliver the requested piece of information accompanied with its quality characteristics in order that the information is verified. Furthermore, the service is informed about the information characteristics in order to be able to adapt its behavior accordingly e.g. inform the user that the context information is possibly inaccurate or imprecise. The time sample or timestamp that describes the production time of the data is also provided. By monitoring the behaviour of the context providers and comparing the quality attributes that each of them gives during registration phase with the ones accompanying the context data, the Context Validation Process estimates the fidelity and the time response. This estimation per each context provider facilitates the optimal context source selection of the next context requests.

3 Context Information Selection Problem

In this section we formulate the optimization problem referring to the selection of context information. The problem objective is to find the preferable context provider to acquire the needed context that results to the maximization of the expected utility. Our model exploits the following input data: First, the selected service profile in order to obtain the properties of context information to be gathered; Second, the upper bound of the economic cost as identified by the CAS customer to be spent for information gathering; Third, the context information provided by the available context sources and the cost for obtaining it. Additionally there are constraints referring to the maximum time and cost for obtaining the required context.

4.1 Formal Problem Description

The market of context information consists of the set of context providers CP_{ij} that sell context information I_{ij} that is of type $i \in (1, N)$, where N is the total number of the available context items, and of the data quality $j \in (1, M_i)$, where M_i denotes the total number of the available quality levels of context items of type i. For every I_{ij} and the corresponding CP_{ij}, the quality properties, depicted in Table 1, are defined.

Table 1. Data Quality Properties of Context Information

Data Quality Property Symbol	Data Quality Property Description
A_{ij}	Accuracy of I_{ij} measured by the same metric units as I_{ij}
$\Pr_{ij}$	Precision of I_{ij} measured by probability
Tr_{ij}	Time refresh of I_{ij} measured by time metrics
Ts_{ij}	Time sample of I_{ij} measured by date and time metrics
$Tresp_{ij}$	Time response of CP_{ij} measured by time metrics
F_{ij}	Fidelity of CP_{ij} measured by probability

Therefore, for each I_{ij} and CP_{ij} $, i \in (1, N)$, $j \in (1, M_i)$ we define the Information Quality Properties Vector $IQPV_{ij} =< A_{ij}, \Pr_{ij}, Tr_{ij}, Ts_{ij} >$ and the Context Provider

Quality Properties Vector $CPPV_{ij} =< Tresp_{ij}, F_{ij} >$. Finally, P_{ij} represents the cost of purchasing information I_{ij}.

During the subscription phase, the CAS customer selects one of the predefined service profiles and enters the parameters that customize the service. The selected service profile requires the acquisition of L specific context items of $\{k_1, k_2, ..., k_L\} \subseteq (1, N)$ types. For each $k_z, z \in (1, L)$ requested context item the Acceptable Properties Vector $APV_{k_z} = \langle A\max_{k_z}, Tr\max_{k_z} \rangle$ is specified as well as the Properties Weight Vector $PWV_{k_z} = \langle a_{k_z}, b_{k_z}, c_{k_z}, d_{k_z}, e_{k_z}, f_{k_z} \rangle$, that weight the importance of quality in terms of accuracy, precision, time validity, producer's time response and fidelity and comply with the following equation: $a_{k_z} + b_{k_z} + c_{k_z} + d_{k_z} + e_{k_z} + f_{k_z} = 1$. Both APV and PWV are determined by statistics produced by the provisioning of the service. Moreover the CAS customer specifies the amount of money that the specific CAS subscription should cost him/her. Based on this value, the maximum cost $P_{\max}$ of purchasing the necessary context information is calculated. Furthermore, the selected service profile suggests the upper bound of the time period $T_{\max}$ for obtaining the information. The problem's input data are summarized in Table 2.

Table 2. Problem's Input Data

	Input Attribute	Input Attribute Symbol
Context Information Market	Information Quality Properties Vector	$IQPV_{ij} =< A_{ij}, \Pr_{ij}, Tr_{ij}, Ts_{ij} >$
	Context Provider Quality Properties Vector	$CPPV_{ij} =< Tresp_{ij}, F_{ij} >$
	Cost of purchasing information I_{ij}	P_{ij}
Service Profile/ User Subscription Data	L specific types of context items to be acquired	$\{k_1, k_2, ..., k_L\} \subseteq (1, N)$
	Acceptable Properties Vector	$APV_{k_z} = \langle A\max_{k_z}, Tr\max_{k_z} \rangle, \forall z \in (1, L)$
	Properties Weight Vector	$PWV_{k_z} = \langle a_{k_z}, b_{k_z}, c_{k_z}, d_{k_z}, e_{k_z}, f_{k_z} \rangle,$ $\forall k_z, z \in (1, L)$
	Maximum cost of purchasing required information	$P_{\max}$
	Maximum time frame for obtaining the information	$T_{\max}$

Having described the parameters of the problem, a formal statement of the problem is as follows: "Given a market of context information consisting of the set of context providers CP_{ij} ($i \in (1, N), j \in (1, M_i)$) that they are selling context information I_{ij}, the corresponding characteristics such as the quality vectors $IQPV_{ij}$ and $CPPV_{ij}$, the cost P_{ij} for obtaining information I_{ij}, a finite number of L context items of distinct types forming the set $\{k_1, k_2, ..., k_L\} \subseteq (1, N)$ that need to be obtained for the CAS that operates under a specific service profile, the constraints imposed by the selected service profile depicted with the vectors $APV_{k_z}, PWV_{k_z} \ \forall k_z, z \in (1, L)$ as well as the maximum acceptable amount of money $P_{\max}$ that can be spent for purchasing the required context information and the maximum acceptable time frame $T_{\max}$ for obtaining the information, the objective is to determine the context producers that will provide the required information so as to maximize the expected utility under the constraints imposed by the service profile."

4.2 Mathematical formulation

The Utility Function $U(I_{ij})$ computes the utilization of information I_{ij} for the specific service and service profile and is comprised by six factors that measure the utility in terms of accuracy, precision, time validity, producer's time response and fidelity. The utility function increases as the context item is more accurate, precise, as it is quicker available to the context consumer, and as it remains valid for a longer period. Regarding accuracy and time refresh, the utility of a context item increases as the properties A_{ij}, Tr_{ij} decrease respectively. The formulation of the factors considers the quality constraints imposed by the selected service profile, namely $A\max_i, Tr\max_i$. The factors take values in [0,1] and are listed below:

- $U_A(I_{ij}) = 1 - \frac{A_{ij}}{A\max_i}$, measuring utility in terms of accuracy.
- $U_{\Pr}(I_{ij}) = \Pr_{ij}$, measuring utility in terms of precision
- $U_T(I_{ij}) = 1 - \frac{T_{current} - Ts_{ij} + Tresp_{ij}}{Tr_{ij}}$ or $U_T(I_{ij}) = 1 - \frac{T_{current} - Ts_{ij}}{\max \bigcup_{k=i,j=1}^{k=i,j=M_i} (T_{current} - Ts_{kj})}$, measuring utility of dynamic and static context information in terms of time validity.
- $U_{Tresp}(I_{ij}) = 1 - \frac{Tresp_{ij}}{T_{\max}}$, measuring utility in terms of producer's time response.
- $U_{Tr}(I_{ij}) = 1 - \frac{Tr_{ij}}{Tr\max_i}$, measuring utility in terms of time refresh.
- $U_F(I_{ij}) = F_{ij}$, measuring utility in terms of producer's fidelity

The utility function of I_{ij} taking into account the ranking of the utility factors as suggested by the selected service profile $PWV_i = \langle a_i, b_i, c_i, d_i, e_i, f_i \rangle$, is formulated as in the following:

$$U(I_{ij}) = a_i U_A(I_{ij}) + b_i U_{\Pr}(I_{ij}) + c_i U_T(I_{ij}) + d_i U_{Tresp}(I_{ij}) + e_i U_{Tr}(I_{ij}) + f_i U_F(I_{ij})$$

In case a quality attribute is not available for the specific context item, the corresponding utility factor equals zero. For example, for the static information the utility function depends only on the time validity of the information, the response time and the fidelity of the producer.

Taking into account the formal description of the problem, the utility function that has to be maximized is the following:

$$UF = \sum_{i=k_1}^{i=k_L} \sum_{j=0}^{j=m} U(I_{ij}) x_{ij} \quad (1)$$

$x_{ij} \in \{0,1\}$ is the decision variable describing if the information of type i is being acquired from the context producer CP_{ij} or not. Moreover, the following constraints should be met in order to find the optimal solution:

$$\sum_{i=k_1}^{i=k_L} \sum_{j=0}^{j=M_i} P_{ij} x_{ij} \le P_{\max} \quad (2)$$

$$\sum_{i=k_1}^{i=k_L} \sum_{j=0}^{j=m} Tresp_{ij} x_{ij} \le T_{\max} \quad (3)$$

$$\forall i \in \{k_1, k_2, ..., k_L\}, \sum_{j=0}^{j=M_i} x_{ij} = 1 \quad (4)$$

$$\forall i \in \{k_1, k_2, ..., k_L\}, \sum_{j=0}^{j=M_i} A_{ij} x_{ij} \le A \max{}_i \quad (5)$$

$$\forall i \in \{k_1, k_2, ..., k_L\}, \sum_{j=0}^{j=M_i} Tr_{ij} x_{ij} \le Tr \max{}_i \quad (6)$$

5 Problem Solution and Results

An important class of combinatorial optimization problems is the Knapsack Problem [11] and its variants. Numerous problems of different fields such as capital budgeting, cargo loading and resource allocation are modeled as a variant of the knapsack problem. The objective of the original Knapsack Problem is to optimize resource allocation, or more precisely, how to distribute a fixed amount of resources among several actions in order to obtain a maximum payoff. Due to the high applicability of it, it has been widely studied and many algorithms for solving it have been proposed. The problem of context provider selection that has been analyzed in the previous section belongs to the category of Multi-Choice Multi-Dimensional Knapsack Problem (MMKP). The definition of MMKP is: There are n groups of items. Group i contains l_i items. Each item has a particular value and it requires m resources. The objective is to pick exactly one item from each group in order to achieve maximum total value of the collected items, subject to the m resource constraints. In the case of the context provider selection problem, each group represents a requested type of info while the items in it represent the available context items. The resource constraints refer to the cost and the required time for obtaining the information. Finally, the value that needs to be maximized is the expected utility of the picked context items.

According to our current work, in order to select the context providers for acquiring the requested information, we apply an exhaustive algorithm. This algorithm computes all possible combinations in order to find the optimal one. Regarding the case that no feasible solution can be found due to constraints (2) or (3), the service policies are considered in order to relax the relevant constraint, namely increase $P_{\max}$ or $T_{\max}$ respectively. In case there is no available context item satisfying (5) or (6) to be obtained for a specific context request for information i, CME selects the one that is expected to provide better satisfaction to the user, namely the one that minimizes the violation of (5) and (6).

In order to demonstrate the proposed approach, we have performed some simulations of a simple context-aware restaurant finder service. Utilizing the exhaustive approach, the context information that is offered by the context providers is evaluated and finally the appropriate ones that best satisfy the customized service's is selected. The CAS scenario considers the user's preferences along with user's location in order to determine the appropriate restaurant for the user to have lunch. In case that the user would like weather conditions, namely temperature information, to be also taken into account, the service retrieves location information of the user and based on this data, it also collects the temperature information. In the Table 3, the properties of the offered location and temperature information are shown.

Table 3. Context Information Market

	Location Information (m)						**Temperature Information (^oC)**					
	I_{11}	I_{12}	I_{13}	I_{14}	I_{15}	I_{16}	I_{21}	I_{22}	I_{23}	I_{24}	I_{25}	I_{26}
P	150	100	70	80	190	20	50	70	80	100	130	40
A	5	100	900	200	1	1000	2	1,5	1	1	0,5	3
Pr (%)	100	100	100	100	100	100	100	100	100	100	100	100
Tr (sec)	10	60	240	100	6	220	10	8	10	5	5	10
Ts (sec)	1	1	1	1	1	1	1	1	1	1	1	1
Tresp(sec)	0,9	2,1	1	0,5	1,2	1,2	0,5	1	1,9	2,5	1	3,5
F (%)	100	100	100	100	100	100	100	100	100	100	100	100

The CAS provider specifies the following weights regarding the gathering of location and temperature information for the specific CAS: PWV_1={0.4,0.05,0.2,0.1,0.2,0.05} and PWV_2= {0.2,0.05,0.3,0.1,0.3,0.05}. Regarding location information, accuracy is of higher importance while regarding temperature information accuracy and time specific properties have almost the same importance. We consider two users A and B: User A wishes that temperature information is taken into account while he/she does not have any requirements regarding the restaurant's location. In this case, the CAS is expected to work well if it retrieves user's location information with low quality as well as the temperature information of the town that hosts the user. On the other hand, User B wishes to find the nearest restaurant that matches his/her preferences. Therefore, for this user the service requires user's location information with greater detail. The two users select the corresponding profiles that match their needs. The imposed constraints are: APV_1= {1000, 240}, APV_2= {3,10} for User A and APV_1= {5,20} for User B. The price ceilings that are defined by each user are P= 100, 170 respectively, while the maximum time response is T_{max}=4sec. Table 2 shows the expected Utility per context item for User A.

Table 2. Utility Estimation for User A

	Location Information						Temperature Information					
	I_{11}	I_{12}	I_{13}	I_{14}	I_{15}	I_{16}	I_{21}	I_{22}	I_{23}	I_{24}	I_{25}	I_{26}
U_A	0,995	0,9	0,1	0,8	0,999	0	0,33	0,5	0,66	0,66	0,833	0
U_{Pr}	1	1	1	1	1	1	1	1	1	1	1	1
U_T	0,91	0,965	0,995	0,995	0,8	0,994	0,95	0,875	0,81	0,5	0,8	0,65
U_{Tresp}	0,775	0,475	0,75	0,875	0,7	0,7	0,875	0,75	0,525	0,375	0,75	0,125
U_{Tr}	0,958	0,75	0	0,583	0,975	0,083	0	0,199	0	0,5	0,5	0
U_F	1	1	1	1	1	1	1	1	1	1	1	1
U	0,949	0,85	0,414	0,823	0,925	0,386	0,539	0,597	0,528	0,570	0,731	0,307

For User A, our approach concludes to the acquisition of information I_{16} and $I_{22.}$ In this case, the quality function (1) is maximized with value 0,386+0,597=0,983 while all constraints are also satisfied. This solution costs 90. If we only considered the constraints APV_1, APV_2 to perform the selection, the information that would be used is I_{16} and $I_{26,}$ achieving total quality 0,949+0,307=1,256 with cost 60. However, this solution would not satisfy constraint (3). For User B, only I_{11} and I_{15} satisfy (5) or (6) constraints, and represent feasible solutions. Our approach decides that the optimal choice is information I_{11} with utility 0,459 and cost 150. The highest utility is 0,79 and is achieved by I_{15}. However, this is not a feasible solution because it costs 190.

The presented service scenario represents the initial evaluation of the proposed model. The achievements from integrating context selection feature in context management frameworks include the reduction of CAS's cost and the imposed network traffic for collecting information. Additionally, it is expected that both service's performance and user's experience from service usage will be improved. The computation complexity of the exhaustive algorithm is $O(M_1 \bullet M_2 \bullet \ldots \bullet M_L)$. If we assume that all groups of information have equal size M context items, the computational complexity is $O\left(M^L\right)$. Therefore, the exhaustive algorithm requires high computation time and can prove critical for the context-aware services that require real time decision making. As the number of context sources and context requests increase, the required time for concluding to the selection increases. For this reason, in our further work we intend to elaborate on algorithms that will optimize the computational complexity. In [12],[13],[14] there are presented some heuristics for solving the MMKP that can be exploited.

6 Conclusions and Future Plans

This paper proposed a novel approach for ad-hoc discovery and organization of the context providers supported by the Context Matching Engine that performs the online evaluation and dynamic selection of context. The description of the proposed model and an initial evaluation of it have been presented. New context sources can be added on-the-fly, while the chosen approach also caters for the provision of different versions of the same information type, with different update frequency, accuracy, format of representation, and price, an especially useful feature for highly mobile users, moving between different heterogeneous environments as well as unsteady context sources. Our plans for future work in this area include the use of heuristics for the optimization of the computational time, the development of an interactive context quality agreement, and an intelligent context quality monitoring mechanism for critical applications. Finally, we intend to demonstrate the benefits of our model utilizing a real-life context-aware service.

References

1. Henricksen, K., Indulska, J.: Modelling and Using Imperfect Context Information. Reasoning and Management associated with the Sixth International Conference on Ubiquitous Computing (UbiComp 2004). Nottingham/England. (September 2004)
2. Hegering, H., Kupper, A., Linnhoff-Popien, C., Reiser, H.: Management Challenges of Context-Aware Services in Ubiquitous Environments. In: M.Brunner, A. Keller (Eds.): Self-Managing Systems. 14th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM 2003). Heidelberg. Germany. (October 2003). Springer-Verlag. Vol. 2867. pp. 246–259
3. Xynogalas, S., Chantzara, M., Sygkouna, I., Vrontis, S., Roussaki, S., Anagnostou, M.: Context Management for the Provision of Adaptive Services to Roaming Users. IEEE Wireless Communications. (April 2004). Vol. 11. No. 2. pp. 40-47
4. Meneses, F.: Context Management for Heterogeneous Administrative Domains. A. Ferscha et al. (eds.) Advances in Pervasive Computing. Austrian Computer Society. Austria. (2004)
5. Dey, A.: Providing Architectural Support for Building Context-Aware Applications. PhD thesis. College of Computing. Georgia Institute of Technology (December 2000)
6. Judd, G., Steenkiste, P.: Providing Contextual Information to Pervasive Computing Applications. In Proceedings of the 1st IEEE International Conference on Pervasive Computing and Communications (PerCom'03)
7. CONTEXT: Active Creation, Delivery and Management of efficient Context Aware Services. IST-2001-38142- CONTEXT. http://context.upc.es
8. Henricksen, K., Indulska, J., Rakotonirainy, A.: Modeling Context Information in Pervasive Computing Systems. In Proceedings of the 1st International Conference on Pervasive Computing, Pervasive 2002. Springer-Verlag. Vol. 2414, pp. 169-180
9. Pipino, L., Lee, Y., Wang, R.: Data Quality Assessment. Communications of the ACM (April 2002). Vol. 45, No. 4. pp. 211-218
10. Buchholz, T., Küpper, A., Schiffers, M.: Quality of Context: What it is and Why We Need It. In Proceedings of the Workshop of the HP OPenView University Association 2003 (HPOVUA 2003). Genevy. (July 2003)
11. Kellerer, H., Pferschy, U., Pisinger, D.: Knapsack Problems. Springer. ISBN: 3-540-40286-1. p. 546.(2004)
12. Khan, S., Li, K., Manning, E, Akbar, M.: Quality Adaptation in a multi-session adaptive multimedia system: model, algorithms and architecture. PhD Thesis, Department of Electronical and Computer Engineering, University of Victoria, Canada. (1998)
13. Moser, M., Jokanovic, D., Shiratori, N.: An Algorithm for the multidimensional multiple-choice knapsack problem. IECE Trans Fundamentals Electron, vol. 80. pp: 582-589. (1997)
14. Hifi, M., Micrafy, M., Sbihi, A. Heuristic algorithms for the multiple-choice multidimensional knapsack problem. Journal of the Operational Research Society. vol. 55. pp:1323-1332. (December 2004)