Efficient Context Management in Context-Aware Environments

Saar De Zutter, Jürgen Slowack, Wesley De Neve, and Rik Van de Walle
Department of Electronics and Information Systems, Multimedia Lab, Ghent University, and
Interdisciplinary Institute for Broadband Technology (IBBT)
Gaston Crommenlaan 8 bus 201
B-9050 Ledeberg-Gent, Belgium
+3293314959

{saar.dezutter, jurgen.slowack, wesley.deneve, rik.vandewalle}@ugent.be

ABSTRACT

Environments in which multimedia applications are used become more complex and gain importance. Users expect applications to be customized to their needs and tailored to their usage environment. Therefore, on the one hand, context environment information needs to be managed in an efficient manner to optimize usage of this valuable information. On the other hand, to improve the collection of this pervasive context information, information required for services in the environment needs to be accounted for. Different technologies exist to describe the usage environment and the interaction with this description environment. In order to optimize management of all this information, our developed efficient context management service links context providers to context consumers and takes into account prior knowledge about these providers and consumers to make decisions. Efficiency of other multimedia services such as context collection and content adaptation services can be enhanced when the discussed methodology is applied.

Keywords
Context-Awareness, Context Management, Efficiency.

1. INTRODUCTION
In the modern multimedia society, different media formats are delivered to all types of devices over a wide range of networks using numerous protocols. Moreover, it is expected that increased accessibility to ubiquitous multimedia systems will become more common. Our daily lives will be surrounded by services aiding us in using all kinds of multimedia applications. When the heterogeneity in the usage environment is not taken into account, it often hinders delivering optimal Quality of Service. To account for the usage environment, on the one hand, descriptions of aspects of the environment need to be collected. On the other hand, the services that adapt their outcome to the usage environment need to define which characteristics are important to be able to deliver optimal quality. By using an efficient context

management service that links the "context that can be collected" to the "context that is needed by multimedia applications" in an optimized manner, collecting, providing, storing, and consuming context descriptions can be done more efficiently.

This paper addresses a novel methodology that describes an efficient context management service. The decisions made by this service are based on the output of context collection services and the input needed by content adaptation services, thereby enhancing their efficiency.

The remainder of this paper is organized as follows. First, related work is provided in Section 2. Section 3 describes what constitutes a context-aware environment. Next, the methodology for efficient context management is given; including the matching, decision taking, and interpretation processes. A real world example showing the use of the discussed methodology can be found in Section 5. Afterwards, an evaluation of the proposed methodology is given. Section 6 summarizes future work and the paper ends with a conclusion.

2. RELATED WORK
Coping with different usage environments is not a new problem and a few solutions have been developed over the years. Efforts are done by various standardization organizations, such as W3C Composite Capabilities/Preference Profile (CC/PP) [5] and MPEG-21 Digital Item Adaptation (DIA) [3] [4]. They provide a flexible manner to express the capabilities of the usage environment and the user preferences. Using CC/PP, profiles can be defined. These profiles can be used to adapt the content to the needs of the consuming device and user, through fine-tuned content selection or transformation. MPEG-21 DIA enables generic adaptation of content based on the environment context.

The decisions taken in most context-aware environments are based on the underlying technology used to collect and consume context [6]. This results in a lack of interoperability. The authors of [2] therefore suggested some important abstractions and support for the field of context-aware computing, enabling designers to more easily build context-aware applications starting from a common basis. However, the problem of regulating the collection of this context and taking efficient decisions concerning context management remains unsolved.

3. CONTEXT-AWARE ENVIRONMENTS
Nowadays, multimedia applications become more and more aware of the environment in which they are used. Our usage

environment is monitored (respecting the privacy rules) to customize the multimedia applications to our preferences and the (varying) environment. The context-awareness expresses itself in two directions. On one side, collected context-related data which describes the usage environment is provided. On the other side, applications show different behaviour according to the context they are provided with. More specifically, from a context's perspective, context collection services are context providers and content adaptation services are context consumers. Such a context-aware environment is pictured in Figure 1. The context management service interfaces with context provider tools at one side of the figure and context consumer tools at the other side. Based on the output of the context collection services and based on the context needed by the content adaptation services, the context management takes automated decisions to optimize efficiency of context collection and content adaptation services.

4. EFFICIENT CONTEXT MANAGEMENT

This section describes a new methodology for incorporating an efficient decision taking process into a context management service that negotiates with context providers and context consumers in order to optimize context management. The different usage environment aspects collected by the providers or needed by the consumers are defined by context axes. Examples of context axes are age category, screen resolution, geo-localization, bandwidth availability, supported media codecs, and available media formats. To set up an efficient context management service, a formalism is developed for indicating along which axes context is provided and consumed. First, multiple context providers and consumers are discussed. Then, based on this formalism, the axes at both sides are matched and a decision taking process is defined. Finally, the decisions are interpreted at provider and consumer sides. For instance, bandwidth, age, preferred language, and supported codecs are collected and bandwidth, age, and resolution can be consumed. However collection of this context is only necessary when it is to be consumed and consumption of context can only happen when it is available.

4.1 Multiple Providers and Multiple Consumers

The context axes that need to be management in our formalism are assigned a unique label ca_h and are contained by the set CA. Set P_k defines which context axes can be collected by context provider k. The cardinality of set P_k is r_k. Set P contains context axis labels from multiple (here q) context providers. Set C_i defines which context axes are needed by context consumer i. The cardinality of set C_i is n_i. Set C contains context axis labels from multiple (here m) context consumers.

$$P_k = \{p_l : p_l \in CA\} \qquad \#P_k = r_k$$

$$P = \bigcup_{k=0}^{q-1} P_k$$

$$C_i = \{c_j : c_j \in CA\} \qquad \#C_i = n_i$$

$$C = \bigcup_{i=0}^{m-1} C_i$$

4.2 Matching Process

To make efficient decisions, the different context axes provided at the one side and needed on the other side need to be matched with each other. This way, we can identify which context axes at provider side are needed by consumers and which context axes needed at consumer side are available through providers. Therefore, each context axis is linked to a value, called "matching value" (noted as $match_h$) and is based on whether the context axis at one side X has a matching context axis on the other side Y. The "matching" function m maps each context axis onto a boolean value: 1 (0) if the context axis from one side is (not) part of the other side.

$$m{:}ca_h \mapsto match_h = \begin{cases} 1 \mid ca_h \in P \cap C \\ 0 \mid ca_h \notin P \cap C \end{cases}$$

$$m(ca_h) = match_h$$

The function m is applied to each context axis in CA, resulting in an extra value for every context axis. This value is 1 if a matching consumer context axis is found for a provider axis and vice versa.
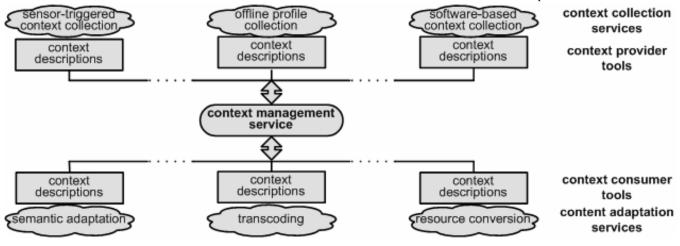

Figure 1. Context management service in a context-aware environment.

For instance, matching value $match_h$ is 1 if context axis h is provided by at least one context provider and used by at least one consumer.

4.3 Decision Taking Process

Based on the matching process, different decisions can be taken to enhance the efficiency of context providers and consumers. From the provider's perspective, decisions on whether or not context is provided are taken based on the need of the consumers. Towards the consumer side, decisions on consuming context are based on the availability of the needed context axes. The "decision" function d maps the context axes onto "decision values" based on the matching values. The interpretation can differ for provider and consumer side. The decisions taken are based on additional information, such as limitation constraints of a context axis h with numerical values provided or consumed by service g (noted as $max_g(|x_h|)$ and $min_g(|x_h|)$). The limitation constraints express the range of context values of a context axis provided or consumed by a certain service. For example, the bandwidth that can be collected lies between 256 kbps and 4 Mbps. Note that x and X are used as placeholders to indicate either context provider or consumer side and g and h are used as placeholders for the index of the provider or consumer and the index of the context axis, respectively. The d function used in the next sections maps to four possible values. The interpretation can differ for provider and consumer side. We extended our formalism so that each context axis of each provider and consumer is linked to a decision value (noted as $decX_{gh}$). Note that, in case context axes are provided (or consumed) by multiple context providers (or consumers), the indices indicate which context provider (or consumer) is providing (or consuming) which context axis.

$$d:(x_h, g) \mapsto decX_{gh} \qquad decX_{gh} \in \{0,1,2,3\}$$

The function d is applied to each context axis of each provider P_k and each consumer C_i, resulting in $\sum_{k=0}^{q-1} r_k + \sum_{i=0}^{m-1} n_i$ extra values.

$$d(p_l, k) = decP_{kl} \qquad d(c_j, i) = decC_{ij}$$

The manner in which the decision values are defined, is explained in the next sections.

4.3.1 Provider Side

Decisions $decP_{kl}$ for all context axes of each context provider k are made based on the matching values. If a matching value $match_l$ is 0, this context axis is not needed by any consumer and thus the decision value is 0. Formally, for each context axis l of each provider k, it is decided that $decP_{kl}$ is 0 if no matching context axis was found by the matching process.

$$match_l == 0 \Rightarrow decP_{kl} = 0$$

If the matching value is 1, the context axis p_l is needed by at least one context consumer and when the limitation constraints of the context axis are taken into account, more efficient decisions can be deducted. If the full range of the context axis p_l provided by context provider k is smaller than the ranges of all consumers c_j consuming this context axis, the decision is 0.

$$match_l = 1 \wedge \forall c_j \in C_i \wedge c_j == p_l :$$
$$min_i(|c_j|) > max_k(|p_l|) \Rightarrow decP_{kl} = 0$$

The only case in which the decisions for context axes have value 1 is the following. If the full range of a context axis p_l provided by context provider k is within the range of at least one consumer c_j consuming this context axis, the decision is 1.

$$match_l = 1 \wedge \exists c_j \in C_i \wedge c_j == p_l :$$
$$min_i(|c_j|) \leq min_k(|p_l|) \wedge max_i(|c_j|) \geq max_k(|p_l|)$$
$$\Rightarrow decP_{kl} = 1$$

The decision for a context axis p_l provided by context provider k has value 2 or 3 when the range of this axis is bigger than the ranges of all matching context consumers c_j consuming this axis or when there is overlap in the range of the provider axis and all matching consumer axes, respectively.

$$match_l = 1 \wedge \forall c_j \in C_i \wedge c_j == p_l : max_i(|c_j|) \leq min_k(|p_l|)$$
$$\Rightarrow decP_{kl} = 2$$
$$match_l = 1 \wedge \forall c_j \in C_i \wedge c_j == p_l :$$
$$((min_i(|c_j|) \geq min_k(|p_l|) \wedge min_i(|c_j|) \leq max_k(|p_l|)) \vee$$
$$(max_i(|c_j|) \leq max_k(|p_l|) \wedge max_i(|c_j|) \geq min_k(|p_l|)))$$
$$\Rightarrow decP_{kl} = 3$$

4.3.2 Consumer Side

Decisions $decC_{ij}$ for all context axes of each context consumer i are made. If a matching value at consumer side is 0, this needed context axis can not be provided and thus the decision value is 0. Formally, for each context axis j of each consumer i, it is decided that $decC_{ij}$ is 0 if no matching context axis was found by the matching process.

$$match_j == 0 \Rightarrow decC_{ij} = 0$$

Also here, if the matching value is 1, the context axis c_j is provided by at least one context provider and the limitation constraints of the context axis can be taken into account to make more efficient decisions. If the ranges of all matching provided context axes p_l are smaller than the range of the context axis c_j that is to be consumed by consumer i, the decision $decC_{ij}$ has value 0.

$$match_j = 1 \wedge \forall p_l \in P \wedge p_l == c_j :$$
$$max_k(|p_l|) < min_i(|c_j|) \Rightarrow decC_{ij} = 0$$

If there is at least one range of a matching context provider axis p_l that is fully covering the range of the context axis c_j that is to be consumed by context consumer i, the decision $decC_{ij}$ is 1.

$$match_j = 1 \wedge \exists p_l \in P_k \wedge p_l == c_j :$$
$$min_k(|p_l|) \leq min_i(|c_j|) \wedge max_k(|p_l|) \geq max_i(|c_j|)$$
$$\Rightarrow decC_{ij} = 1$$

The decision for a consumer context axis has value 2 or 3 when the range of this axis c_j is lower than the ranges of all matching context consumers p_l and when there is overlap in the range of the consumer axis and all matching provider axes, respectively.

$$match_j = 1 \land \forall p_l \in P_k \land p_l == c_j : \min_k(|p_l|) \geq \max_i(|c_j|)$$
$$\Rightarrow decC_{ij} = 2$$
$$match_j = 1 \land \forall p_l \in P_k \land p_l == c_j :$$
$$((\min_i(|p_j|) \geq \min_k(|c_l|) \land \min_i(|p_j|) \leq \max_k(|c_l|)) \lor$$
$$(\max_i(|p_j|) \leq \max_k(|c_l|) \land \max_i(|p_j|) \geq \min_k(|c_l|)))$$
$$\Rightarrow decC_{ij} = 3$$

4.4 Interpretation Process

Based on the formalism defined above, a concrete interpretation can be given to the taken decisions. This section applies the interpretation process to context collection services and content adaptation services in context-aware environments at provider side and consumer side, respectively. The interpretation process occurs on two levels: at the context axis level and at the context provider/consumer level.

4.4.1 Provider Side

If $decP_{kl}$ has value 0, context axis l of context provider k does not need to be provided. Moreover, when interpreting the decisions at the provider level, if all decision values of a context provider k are 0, none of the context axes of this provider are needed. Thus, each context provider k for which the decision values add up to 0 can be fully eliminated from the context management system since none of the context values they provide are needed by any adaptation service.

$$\sum_{l=0}^{r_k-1} decP_{kl} = 0 \Rightarrow \text{eliminate provider } k$$

If $decP_{kl}$ has value 1, 2, or 3, the corresponding context needs to be provided. However, in case $decP_{kl}$ equals 3, only a partial range of values needs to be provided, i.e., within the range of limitations of the consuming content adaptation service(s). If $decP_{kl}$ equals 2, the provided context values can be delimited to the maximum of the range of needed context values.

4.4.2 Consumer Side

If $decC_{ij}$ has value 0, adaptation i to context axis j is not possible, since the needed context values are not provided or exceed the range of the provided context values. Moreover, when interpreting the decisions at the consumer level, if at least one of the decision values of a context consumer i is 0, the adaptation using consumer i does not have access to all needed information and can therefore not be performed. The full range of adaptation possibilities can be exploited for all content adaptation services i for which the sum of the decision values that are 1 equals the amount of needed context axes.

$$\sum_{j=0}^{n_i-1} decC_{ij} = n_i \land decC_{ij} == 1 \Rightarrow \text{full consumption } i$$

If $decC_{ij}$ equals 1, consumption of this context axis is possible. In case $decC_{ij}$ has value 2, the adaptation i to context axis j will be inferior, meaning that the content is adapted based on constraints which are stricter than the ones provided. If $decC_{ij}$ equals 3, the adaptation needs to be restricted to the range of provided values.

5. EVALUATION OF REAL WORLD EXAMPLE

In a real world example, a context provider could describe the context it is able to collect by using MPEG-21 DIA Configuration (DIAC) [4]. Content adaptation services can also use DIAC to describe the needed context to adapt the content to. The limitations on the context values can be defined using MPEG-21 DIA Universal Constraint Descriptions [3]. The context axes (and corresponding decision values) that can be provided by 3 context providers (i.e., sensor-triggered service (P_1), offline profile service (P_2), and software-based services (P_3)) are shown below on the left and the context axis needed by 2 context consumers (i.e., semantic adaptation service (C_1) and transcoding service (C_2)) can be found on the right. When our context management methodology is applied, it becomes clear that the last context provider is not needed and the language does not need to be collected. It can also be decided that transcoding is not possible since the needed context values are not available. Semantic adaptation can be applied when the provided context values are within range of the needed values.

$$\begin{cases} P_1(\text{bandwidth},1) \\ P_2(\text{age},1) \quad\quad (\text{language},0) \\ P_3(\text{codecs},0) \end{cases} \begin{cases} C_1(\text{bandwidth},3) \ (\text{age},1) \\ C_2(\text{resolution},0) \end{cases}$$

To evaluate the proposed methodology, it is necessary to measure the amount of data that is sent in a context-aware environment and to compare this to the amount of data sent when the proposed context management service is used. It is also important to take into account the extra overhead the context management service introduces. To evaluate the performance gain in the amount of resources needed to apply the proposed methodology, we analyzed the complexity and efficiency of the algorithm behind the methodology by means of asymptotic analysis and notation [1]. The times $Tmatch$, $Tdec$, and $Tinterp$ needed to define the matching values of all context axes and decision values for every context axis of all providers and consumers and the corresponding interpretation are specified in Table 1. There are $\sum_{k=0}^{q-1} r_k + \sum_{i=0}^{m-1} n_i$ decision values, this sum is below referred to as $p+c$. As can be seen, the algorithm has order of $(p+c)^2$ time complexity. The calculation time required for this methodology increases quadratic with the amount of context axes provided and consumed in the context-aware environment. The whole process is often done once at setup of the environment or is done gradually when new providers and consumers are added. To make it worthwhile to spend time $Tmeth$ on managing context providers and consumers, the gain of exchanging less context values as a consequence of the interpretation process must be higher than $Tmeth$. Therefore, we have defined the asymptotic behavior of time spent on exchanging n context values without ($Texch$) and with ($Texch\_meth$) the use of our methodology. It can be seen that there is an overhead in data to be exchanged to set up the context management service to be able to define the interpretation values and to inform the respective context providers and consumers. Once this initial start-up phase is done, only 50% on average of the context values needs to be exchanged. This means that from the moment half of the amount of exchanged context values is in the same order as the quadrate of

the amount of providers and consumers, using the context management service is more efficient. The asymptotic analysis indicates the average time needed in case of a lot of context axes, not the best case time nor the worst case time. The asymptotic lower, respectively upper, bound is $\Omega(1)$, resp. $O(n)$. The more providers provide context values not needed by consumers and vice versa, the more performance gain we can become.

Table 1. Evaluation of proposed methodology

	Θ
Matching Process *Tmatch*	$\Theta(p+c)^2$
Decision Taken Process *Tdec*	$\Theta(p+c)^2$
Interpretation Process *Tinterp*	$\Theta(p+c)$
Methodology *Tmeth = Tmatch + Tdec + Tinterp*	$\Theta(p+c)^2$
Exchange *n* context values without use of proposed methodology *Texch*	$\Theta(n)$
Exchange *n* context values based on proposed methodology *Texch_meth*	$\Theta(n/2)$

6. FUTURE WORK

The methodology discussed in this paper can be further developed to include intelligence. Based on extra information about the usage environment and services that are sensitive to this environment, the matching and decisions taking processes can be enhanced. In this paper, matching is done on an "availability/need" basis; enhanced matching can, for example, be based on geo-localization information of the different services. On top of limitation constraints on the different context axes, other information can be used as input to the decision taking process, e.g., optimization constraints, time-based information, and dependencies between context axes. Focusing on specifically tailored context-aware environments, more advanced interpretation systems can be built.

It is worth investigating how this methodology enhances efficiency when using different standards for context description, such as CC/PP and MPEG-21 DIA, in context-aware environments.

7. CONCLUSION

This paper describes a methodology for including efficiency in context management services. The decision process of the developed service is based on the context available from context providers and the context needed by context consumers.

Applying the methodology in context-aware environments results in more efficient context collection and content adaptation services. Already existing adaptation decision taking processes can be fine-tuned when the output of the efficient context management service is taken into account. Context collection can be improved, since less redundant context information traffic is needed.

9. REFERENCES

[1] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C., *Introduction to Algorithms, Second Edition*. MIT Press and McGraw-Hill, 2001, 3–122.

[2] Dey, A. K. and Abowd, G. D. The Context Toolkit: Aiding the Development of Context-Aware Applications. In *Proceedings of the Workshop on Software Engineering for Wearable and Pervasive Computing* (Limerick, Ireland, June 6, 2000). ACM Press, New York, NY, 2000.

[3] Timmerer, C., Devillers, S., and Vetro, A. *Digital Item Adaptation – Coding Format Independence*. In The MPEG-21 Book. John Wiley & Sons Ltd, New Jersey, 2006, 283-332.

[4] Vetro, A., Timmerer, C., and Devillers, S., *Digital Item Adaptation – Tools for Universal Multimedia Access*. In The MPEG-21 Book. John Wiley & Sons Ltd, New Jersey, 2006, 243-282.

[5] World Wide Web Consortium Recommendation. *Composite Capability / Preference Profile (CC/PP): Structure and Vocabularies*. W3C Technical Report, 2004, http://www.w3.org/TR/CCPP-struct-vocab/.

[6] Zimmerman, A., Specht, M., and Lorenz, A. Personalization and Context Management. In *User Modeling and User Adapted Interaction*. Kluwer Academic Publishers, Hingham, MA, USA, 2005, 15, 3-4, 275-302.