A Model-Based Framework for Adaptive Resource Management in Mobile Augmented Reality System

Mykola Tkachuk [1], Oleksii Vekshyn [1], Rustam Gamzayev [1]

[1] National Technical University "Kharkiv Polytechnic Institute", Frunze str., 21, Kharkov, Ukraine
tka@kpi.kharkov.ua, alexeyvekshin@gmail.com, rustam.gamzayev@gmail.com

Abstract. A 3-level modeling framework to adaptive resource management in mobile augmented reality systems (MARS) is proposed, which is based on comprehensive data structuring and analyzing of their specific hard- and software features. At the conceptual modeling level an ontological specification of MARS resources is constructed, at the logical modeling level a case-based reasoning algorithm is elaborated, and as a physical model the reference software architecture is designed. This approach was successfully tested to solve the task to adaptive management of image resolution on mobile device (MD) according to changes of computational loading that finally enabled better video stream quality in MARS.

Keywords. model, adaptation, resource management, augmented reality, case-based reasoning.

Key Terms. ModelBasedSoftwareDevelopmentMethodology, Model, SoftwareSystem

1 Introduction

Nowadays, mobile information systems become more and more popular. One of the most complex and dynamically grown type of these systems are mobile augmented reality systems (MARS) [1]. Such systems require more hardware resources than standard mobile applications (e.g. social network clients, instant messengers etc.), and this fact leads to supporting problems of different devices such as mobile phones and tablets. One of the possible solutions is an execution of complex business logic on the server side, where computational capabilities are higher than on the mobile client side; but on the other hand this, could lead to problems with application response time and energy efficiency because of more intensive usage of wireless networking technologies. One of the overriding trends in software development is using of complex systems construction principles, especially cybernetic adaptive control schemes for software components control including appropriate decision making models and quality evaluation metrics [2]. These approaches are also useful in case of mobile infor-

ICTERI 2016, Kyiv, Ukraine, June 21-24, 2016

mation systems development, in particular for MARS. Such systems development requires effective utilization of restricted mobile device (MD) resources and on the other hand implementation of complex real-time algorithms.

In this paper we propose an adaptive model-based framework for resource management in MARS. Additionally a prototype implementation of adaptive MARS is presented and series of experiments with proposed framework and MARS are provided. These experiments highlight positive effect of using adaptive approaches in MARS development.

This paper is structured in the following way: Section 2 depicts briefly some modern trends in this research domain, with respect to some adaptation issues; Section 3 provides main concept of adaptive model-based framework for resource management in MARS; Section 4 presents some ARS domain modeling issues like ontological specifications; in Section 5 the algorithmic model for adaptive resources management and metrics are given; in Section 6 presents proof of concept, introduces software prototype and experimental results. Finally, a short outlook on the result achieved and some future work is presented.

2 Related Work in the Domain of Adaptive Augmented Reality System Development

An Augmented reality (AR) is a representational form for a real physical environment, which is extended by adding of computer-generated data [3]. AR registers physical objects in three dimensions and combines them with the virtual ones. Apart from virtual reality, AR combines models of objects from real world with additional information, but virtual reality completely replaces the real world with the virtual one.

Currently AR supports several data sources: two-dimensional markers; data received from GPS-modules (Global Positioning System) [4] and from build-in gyroscopes. Additionally ARS could use some modern technologies like images recognition without any markers and GPS data [5].

With respect to supported data source, it is possible to construct some MARS classification, which is presented in Fig. 1. There are four “top-level” types of MARS; each of these types has some features and requirements to MD performance. Marker-based MARS operates with special markers, which stores some data and links to additional information. Non-marker based MARS are more complex types of these systems, it based on image recognition algorithms and requires more resources to find and recognize free-form objects in an input image. Geolocation MARS uses build-in GPS sensors in order to get information about real environment and augment it with some virtual data. Finally, infrared-sensors based MARS operates with some infrared sensors, which is able to detect object and moves in real environment, subsequently this class of MARS is very useful in entertainment and simulators. In scope of this research we are focused on Markes-based MARS, because this class of MARS has the lowest complexity, easy to implement and does not require any additional equipment apart from MD.

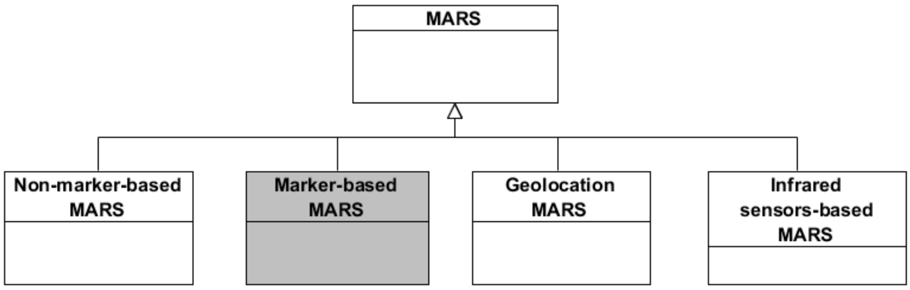

Fig. 1. Classification of MARS types

In terms of software development there are some modern software frameworks to develop MARS: Metaio Mobile SDK (Software Development Kit) [6], D'Fusion Mobile [7] and Qualcomm [8].

Nowadays, software adaptation is one of the common trends in modern software engineering (see, e.g., in [9]), and especially in mobile application development. There are several approaches to adaptation in mobile systems, some of them are represented in projects like Q-CAD (QoS and Context Aware Discovery), MADAM (Mobility and Adap-tation Enabling Middleware), IST-MUSIC (Self-Adapting Applications for Mobile Users in Ubiquitous Computing Environments) [10], CloudRidAR (A Cloud-based Architecture for Mobile Augmented Reality) [11] and researches like Elastic Application Model [12] and resource management in mobile cloud computing [13].

Q-CAD is a resource discovery framework which enables mobile applications to dis-cover and to select resources best satisfied the user's needs. MADAM and ITS-MUSIC frameworks provide model-driven development approach enabling to assemble applications through a recursive composition process. In this case, variability is achieved by plugging into the same component type different component's implementation with similar functional behavior [10]. In [14], a new approach to the composition of mismatching components in context-aware systems is introduced. CloudRidAR is a cloud-based framework to MARS development which provides development facilities to construct MARS using all advantages of cloud computing and code offloading, but on the other hand this framework forces developer to use quite complex design solutions (e.g. could computing, workflows, etc) [11].

Elastic Application Model [12] based on code offloading, but flexible application architecture and models are built only on the server side called "weblets" and MD hosts only simple client application, which is connected to these weblets. Resource management in mobile cloud computing [13] addressed to cloud structure, code offloading and energy efficiency but not to runtime application adaptation.

To sum up related work investigation we can conclude that none of existing approached doesn't provide complete model based framework to adaptive resource management in MARS, but take into account particular aspects of this problem.

In the Fig. 2 and in the Fig. 3, the interface of marker-based MARS is presented. This MARS detects source object in the input video stream recognize it, obtain 3D graphical model of this gear and finally augment source image with this model.

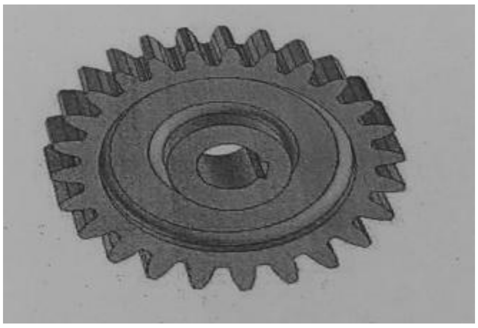

Fig. 2. Not augmented image

Fig. 2 demonstrates interface of the marker-based MARS with not augmented image. In current state MARS is ready to analyze source image and search for markers.

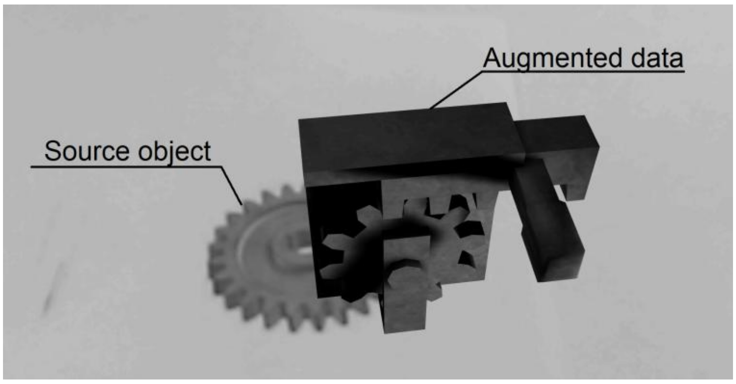

Fig. 3. Example image, augmented by marker-based MARS

In the Fig. 3 result image is shown, MARS finds marker in the source image (in this case – gear), obtains related data and augments source image with this data. In this case, result of augmentation is the gear’s 3D model which is presented over sources gear image.

Marker-based MARS [15] uses different marker recognition approaches, such AS QR-codes [16] or barcodes [17]. One of the typical examples of such MARS is QR Droid application [18].

3 Adaptive Model-based Framework for Resource Management

Taking into account the results of the provided analysis and based on the understanding of modern trends in the domain of adaptive MARS-development (see Section 2), we can conclude that it is necessary to elaborate a complex model-based framework for adaptive resource management in MARS. This assumption is completely corresponded with such well-proved and recognized approaches in modern software development as model-driven development (MDD) and model-driven architecture (MDA) [19]. Last time these issues are already discussed intensively in a lot of publications about resource management in distributed real-time systems [20], in SOA- and cloud-centered applications [13, 21, 23], but there is a lack on such work in the domain of MARS development. That is why we propose to construct such model-based framework in the following way:

- to elaborate a domain model specify to describe all hard- and soft-ware resources to be analyzed and considered in any adaptive procedure in MARS, and such a vision services as a conceptual level in the proposed framework;
- to propose some algorithmic approaches to manage these resources in adaptive mode, and this vision about the resources in MARS can be considered as a logical modeling level in our framework;
- to develop a reference software architecture to implement a logical model of MARS with appropriate components and interfaces, and such architecting has to be recognized as physical modeling level in this framework.

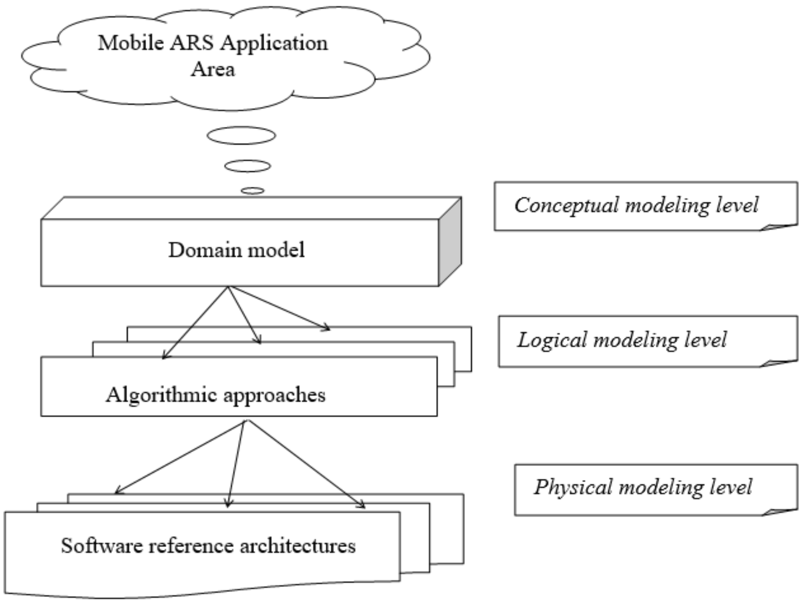

Fig. 4. 3-level modeling framework for MARS adaptation

It is to note that according to this model-based vision about resource management in MARS for the one and the same domain model a lot of different algorithmic approaches can be elaborated, and for any such an approach several reference software architectures might be implemented.

The interacting between these abstraction levels which build the proposed model-based framework is shown in Fig. 4.

In this way, it is possible to elaborate a lot of knowledge-oriented and reusable algorithm-centered solutions and software components, which support the adaptive approach to resource management in MARS. More detailed these issues are considered below.

4 Ontological Specifications for MARS Domain Modeling

Taking into account the 3-level modeling framework for adaptive resource management in MARS proposed in previous Section 3, it is needed to elaborate the domain model for this purpose. This model should represent all relevant hard- and software capabilities of MD which can be considered as adaptable parameters in an appropriate algorithmic modeling approach.

Ontology models are widely used to represent relationships between concepts in some application domain, and they can be applied for different purposes in software engineering, e.g.:

1. for information sharing between human and machines in Semantic Web applications [23];
2. for natural language processing, and knowledge engineering,
3. in software product line engineering [24], etc.

For example, [25] represent the domain model including 4 ontologies like User, Service, Environment and Device. It describes general relationships that occur during ARS development. In [26] ontology is used to build platform for educational institutions that could be used for rapid ARS development. In [27] the ontological specifications are used to connect together the knowledge about the users, environment, and user aims in the given application domain (museum).

Most of models mentioned above describe different MARS components taking into account some static resource allocation, and they do not represent system features needed for adaptive resource management. To close this gap in our approach the new MARS model was created using the OWL notation, and the OWLGrEd tool is used for this purpose [28], which provides the UML-like graphical editor. In Fig. 5, the proposed ontological domain model for MARS is presented.

There are following main entities included in our domain model

1. *Augmented Reality Application*: is an application that provides a technology to analyze elements in the real physical environment, and to extend them with additional virtual objects or with additional information.

2. *ACU (Adaptive Control Unit)*: this is control management element that is responsible to adapt system to the mobile device. It could use different techniques to optimize application, e.g. to change display resolution.
3. *Mobile Device*: this is a small device (iPhone, PDA, notebook, tablet etc.), with restricted amount of system resources, which does not have permanent access to the power sources, and which uses wireless communication technologies. For resource adaptations, the following device parts (model entities) could be used: *Screen* – by changing screen resolution performance of the device could be changed; *RAM* – depending on the allocated memory different application modes might be used; *Battery* - usually a charged level is used as a main parameter to switch device into power save mode; modern *CPU* could change frequency and switch its operation mode.
4. *AR Object Recognizer*: is a module to analyze and to extend extracted physical objects with additional or virtual information.

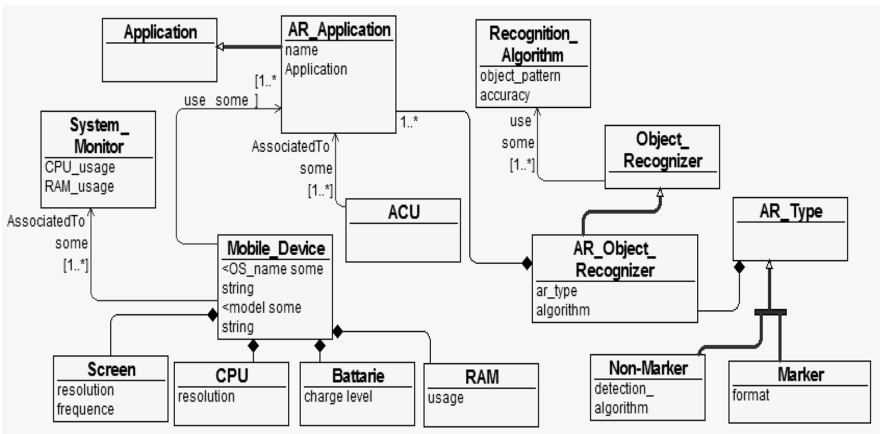

Fig. 5 Ontological domain model for resource adaptation in MARS

Below this domain model is used to elaborate an algorithmic approach to adaptive resource management in MARS.

5 Algorithmic Model of Adaptive Resource Management in MARS

According to the proposed adaptive model-based framework (see Section 3) at its logical level an algorithmic model to resources management has be constructed with respect to specific hard- and software characteristics of MD which is used to get client applications running within a given MARS.

5.1 Formal Definition

In order to formalize the proposed framework to find adaptive solutions for resource management we can use an algorithmic modeling approach [29], and the appropriate algorithmic model AM can be defined as the following tuple

$$AM = \langle Workflow(Methods), InfoBase, Metrics \rangle, \quad (1)$$

where $Workflow(Methods)$ are some algorithms which implement the given methods $(Methods)$, $InfoBase$ is an information base to be used for these methods, and $Metrics$ is a collection of metrics to assess a quality of adaptation process in mobile ARS. The choice of a set of adaptation methods in (1) depend on specific features of ARS's resources, which should be managed, and one of such possible way will be considered in the next subsection. A set of metrics in (1) also has to be defined taking into account the appropriate hard- and software properties of MD which is used in a target ARS (see e.g. in [10, 15]).

5.2 Case-based reasoning (CBR) within an algorithmic model

Taking into account a complex and weak-formalized character of ARS – functioning, namely: parallel and multi-threaded calculation processes, turbulence loading on MD, permanent changes on number of users etc., it is reasonable to use so-called soft calculation methods [30]: neuronal net technologies, fuzzy-logic methods, generic algorithms, case-based reasoning (CBR) and some others. In particular, exactly CBR-methods can be considered as an effective way to develop decision-making procedures for management of complex software system (see e.g. in [31-33]. According to this statement the collection of Methods in formula (1) can be specified in the following form

$$Methods = (NNM, kNNM, kwNNM), \quad (2)$$

where NNM is a Nearest Neighbor Method, kNNM is a k-Nearest Neighbors Method, and kwNNM is k-weighted Nearest Neighbors Method [31].

The main idea of all CBR-methods is that any new problem occurred in some application domain can be resolved using already existing solution for the similar situation (called precedent or case). The several CBR differ each other in a search algorithm to find an appropriate precedent in the given case - database. For this purpose, it is also important to elaborate an adequate description for the precedent's representation, which reflects all relevant issues of ARS functionality.

5.3 Information base for CBR-method

Corresponding to formula (1), InfoBase is an information base which is used to apply the CBR-methods defined in (2). It includes a set of precedents, and any such precedent can be defined in the following way

$$c=(\vec{p},\vec{s}),\tag{3}$$

where $\vec{p}$ is a vector of parameters to characterize a given problem situation, $\vec{s}$ is a vector of parameters to represent an appropriate solution for this problem.

Taking into account the hardware - and software issues of MD which is included in ARS, vector $\vec{p}$ can be given as:

$$\vec{p}=(CPU, RAM, BAT, RES, FPS),\tag{4}$$

where CPU is a current level of processor loading (in %), RAM is a current level of RAM usage, BAT is a current level of battery charging; RES is a number of possible screen resolution modes in MD, FPS is a measure of a screen refresh rate.

Vector $\vec{s}$ in formula (2) can be represented as the tuple

$$\vec{s} = (Width, Hight),\tag{5}$$

where Width and Height are respectively a width and a height of a video frame on MD.

5.4 Metrics for ARS resource estimation

In [15] is mentioned that a performance of a ARS-client application is depend on its screen resolution, and accordingly to this reason a number of frame per second (FPS) can be used as one of the metrics from its set Metrics defined in formula (1). This factor is depend on some parameters: on power of MD processor, on size of its RAM, on screen resolution of MD, and on screen resolution of video-camera.

Therefore, a collection of metrics *Metrics* in (1) has the following definition

$$\vec{s} = (T, R),\tag{6}$$

where T is a metric to estimate a number of frame per second, P is a metric to measure a MD total pRoductivity (R).

A value of metric T can be calculated using the standard function Count(), namely:

$$T = Count(FPS)\tag{7}$$

A value of metric R (named below as a productivity index) defines a MD performance ratio, which is dimensionless parameter and it can be defined as following

$$R = w_c \frac{CPU_{cur}}{CPU_{total}} + w_r \frac{RAM_{cur}}{RAM_{total}} + w_b \frac{BAT_{cur}}{BAT_{total}}\tag{8}$$

where CPUcur is a current MD processor loading ratio (in %); RAMcur is a current RAM usage ratio (in Kb); BATcur is a current battery charging ratio (in Ah); CPUtotal, RAMtotal , BATtotal are respectively the nominal values of the given parameters; w_c, w_r, w_b are some weighting coefficients for these parameters, and the following condition must be fulfilled $w_c + w_r + w_b = 1$. In this work we have defined the following value ranges for the index R: it is critical if $R \geq 0.95$; it is high if $0.6 \leq R < 0.95$; it is normal if $0.25 \leq R < 0.6$; and it is low if $0 \leq R \leq 0.25$.

Therefore, the metrics defined in formula (6) – (8) allow us to estimate the computational resources of an appropriate MD which is used in a target ARS with respect to our final goal: to provide an adaptive recourses management in this ARS.

6 Prove of Concept: Software Prototype and Experimental Results

6.1 Software prototype design and implementation

In order to prove efficiency of the proposed approach the MARS prototype with integrated ACU has been developed. The main purpose of developed MARS is to recognize marker on a cinema poster, search information about this cinema in an appropriate database and augment source video stream with this additional information at real-time mode. Such MARS with integrated ACU which analyzes environmental parameters and adopts frame size with respect to MD current state and resources utilization rate. In the Fig. 6 is presented adaptive MARS functioning algorithm in form of UML activity diagram.

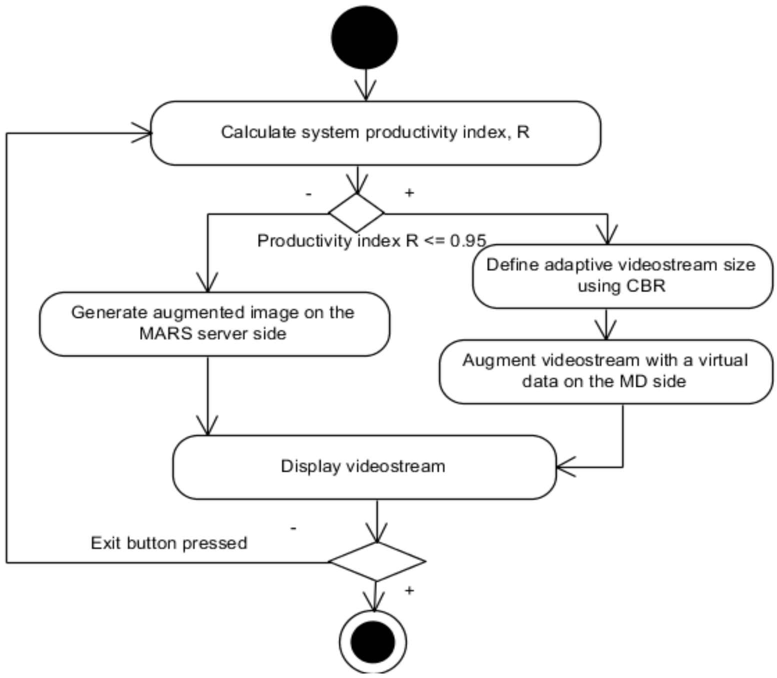

Fig. 6. Adaptive MARS functioning algorithm

Initial activity in this algorithm is – calculation of a productivity index. If this index is less than 0.95 ($R \leq 0.95$) it is possible to augment data on the MD side, so the next steps are to define adaptive video stream size, using CBR method, and augment source video stream with a virtual data on a MD side, and finally – display augmented video stream to user. If index $R > 0.95$ it is not possible to augment data on the MD side and in this case it is necessary to use external resources to augment image from input video stream (e.g. server in client-server MARS) and show result to user. This activity could be interrupted by user's event (pressing exit button).

Presented algorithm takes into account some code offloading possibility (in case of quite high computation load on a MD), subsequently it is useful to select three-tier software architecture to implement adaptive MARS prototype. Such architecture allows us to implement recognition component on the server side in order to offload some logic from MD in case of critical computational load. Another plus point of this architecture is a possibility to deploy centralized precedent database on the server side and distribute this precedents among different MDs.

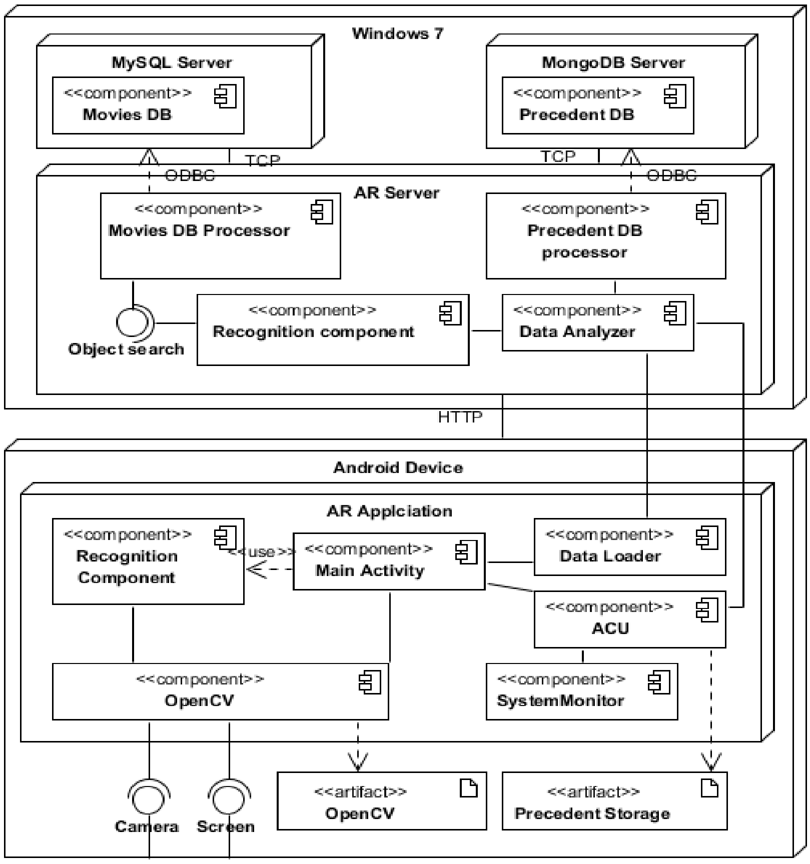

Fig. 7. Adaptive MARS functioning algorithm

In the Fig. 7 presented three-tier component software architecture of MARS in form of UML deployment diagram. This architecture provides few crucial advantages such as: high scalability, data processing security, lower resource requirements for clients MD.

Client MARS application implemented with Android platform [34], using embedded Berkeley DB [35] and OpenCV library [36]. To develop server-side application PHP programming language [37], Apache web-server [38], MySQL [39] and MongoDB [40] have been selected.

ACU on a MD node (Android Device) contains the following components: Precedent-Storage – local precedent DB, SystemMonitor – the component which provides data regarding current state of MD and ACU – the component, which implements CBR methods on the mobile client side. On the server node into adaptation process are involved the following components: Precedent DB Processor is the accessor component for centralized Precedents DB, and Data Analyzer is the component that implements movie's additional data search by input images hash-code.

With respect to the proposed architectural solution, three databases have been implemented: 1) local embedded precedents DB (PrecedentStorage), this DB is used by ACU; 2) the remote centralized DB (Precedent DB) to store all precedents, all local DBs are synchronized with this one; 3) movies DB (Movies DB), this DB stores information about movies, which are handled in MARS prototype. Conceptual data model of the Precedent DB is presented in the Fig. 8 as the UML class diagram. This data model takes into account the following entities: Device, Precedent, Param, ListPrecedent, Platform, TypeParameter. Therefore, it allows handling data, required by CBR method in our domain.

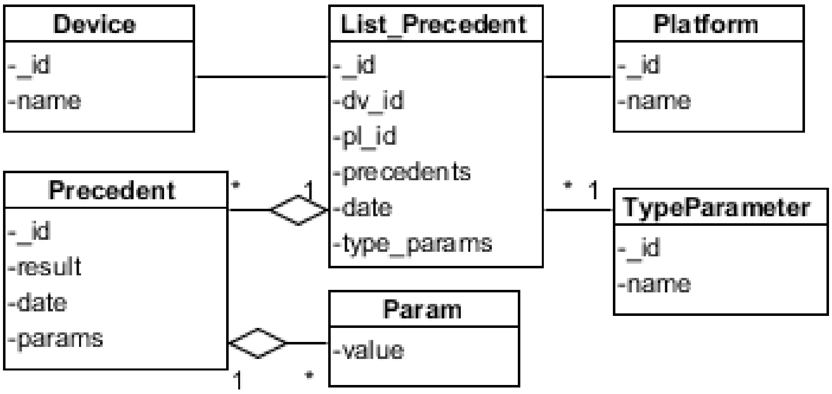

Fig. 8. Conceptual data model of the Precedent DB

To develop this database we have selected non-relational (NoSQL) database management system MongoDB [40]. This DBMS provides high-speed data processing and stores the appropriate information in object-oriented form.

6.2 Estimation results and their analysis

In order to estimate an efficiency of the implemented MARS prototype the experiments have been performed using the following scheme: 1) selection of mobile devices for experiments; 2) precedents DB generation; 3) run MARS prototype with different operating modes for ACU (with enabled and disabled ACU).

To test MARS prototype two types of MD were selected, the detailed characteristics of these devices are presented in the Table 1.

Table 1. Technical characteristics of testing MD

Device name	Processor	RAM	Maximal resolution
Nexus 7	Qualcomm APQ8064 (1.5 GHz)	2 GB RAM	1920x1080
Fly IQ4416	MT6572 (1.3 GHz)	512 MB RAM	800x600

In the Fig. 9 presented example of tuple, which describe particular precedents included in the precedents DB.

№	Result	CPU	RAM	BATTERY	RESOLUTION	FPS
1	640x480	0.7	0.7	0.4	800x600	13
2	800x600	0.2	0.5	0.9	640x480	17
3	1024x768	0.3	0.8	0.5	1920x1080	2

Fig. 9. Example of precedents tuple in precedents DB

Two series of experiments have been provided with these mobile devices. The first experiment has been provided with disabled ACU (i.e. without adaptation), and the second one with enabled ACU. During these experiments the image resolution of MD screen in case of different values of productivity index (see equation 8) had been measured. In this experiment we take into account two intervals from normal and high ranges of the productivity index: $0.4 \le R < 0.6$ and $0.6 < R \le 0.8$.The results of these experiments are presented in Table 2 and Table 3 respectively.

Table 2. Experimental results in case of disabled ACU

Mobile device	Resolution	T (FPS) for $0.4 \le R < 0.6$	T (FPS) for $0.6 < R \le 0.8$
Nexus 7	640x480	15	13
Nexus 7	800x600	14	10
Fly IQ4416	800x600	10	9
Fly IQ4416	480x320	15	14

The data from the Table 2 show that in case of the fixed image resolution on MD, and if the value of productivity index R is increased: from the values range [0.4; 0.6] to the range [0.6; 0.8], then the T(FPS) metric value is decreased, namely, these values are placed in interval [9, 14]. In other words, the maximum value's difference T(FPS) is about 35.7% apart from difference 33,3% for case of [10, 15] ($0.4 \le R < 0.6$). The reason of such trend in this experiment is the disabled mode of ACU.

Table 3. Experimental results in case of enabled ACU

Mobile device	Resolution	T (FPS) $0.4 \le R < 0.6$	Resolution	T (FPS) $0.6 < R \le 0.8$
Nexus 7	1024x768	11	800x600	13
Nexus 7	800x600	14	640x480	16
Fly IQ4416	640x480	13	640x480	14
Fly IQ4416	800x600	9	640x480	13

Table 3 represents experiment results with enabled ACU. ACU component monitors computational load on a mobile device and in case of its increasing correct image resolution; such correction leads to stabilization of T (FPS) metric: this metric changes only in 18.75%. Additionally, in this experiment value of the maximal difference for T (FPS) in case of $0.4 \le R < 0.6$ is 37.5%. That is why we can make conclusion that the proposed adaptive resource management approach enables better video stream quality for mobile device in MARS.

7 Conclusions and Future Work

In this paper we have presented the model-based framework to adaptive resource management in mobile augmented reality systems (MARS), which is based on the 3-level data structuring and analyzing of their specific hard- and software features. At the conceptual modeling level the appropriate ontological specification of MARS resources was constructed, at the logical modeling level a case-based reasoning approach is utilized, and as a physical model to implement an adaptive resource management in MARS the 3-level reference software architecture is elaborated. This approach was successfully applied in order to solve the task to adaptive management of screen image resolution on mobile device according to changes of its computational loading that finally enabled better video stream quality in MARS.

In future we are going to extend a collection of decision search methodologies in order to improve an adaptation process and compare its efficiency with case-based reasoning approach implementation. Besides that is it supposed to develop a more sophisticated adaptive MARS domain model with wider amount of input and output parameters, which should enable a more configure options in the proposed model-based adaptive resource management framework.

8 References

1. Lopez, H., Navarro, A., Relano, J.: An Analysis of Augmented Reality Systems. In: Proceedings of the 2010 Fifth International Multi-conference on Computing in the Global Information Technology (ICCGI '10), pp. 245--250 (2010)
2. Joao W. C., Scott D. M., Kai-Yuan C., Aditya P. M.: Software Cybernetics. Encyclopedia of Computer Science and Engineering. WILEY (2008).
3. Furth, B.: Handbook of Augmented Reality. Springer-Verlag, New York (2011)
4. Official U.S. Government information about the Global Positioning System (GPS) and related topics, http://www.gps.gov
5. MacWilliams, A.: Decentralized Adaptive Architecture for Ubiquitous Augmented Reality Systems. PhD thesis, Institut fur Informatik der Technischen Universitat Munchen (2005)
6. Official Web-site of Metaio Mobile SDK, http://www.metaio.com/software/mobile-sdk/
7. Official Web-site of D'Fusion Mobile project, http://www.t-immersion.com/products/dfusion-suite/dfusion-mobile
8. Official Web-site of Qualcomm AR SDK, http://www.qualcomm.com/solutions/augmented-reality
9. Kell, S.: A Survey of Practical Software Adaptation Techniques. In: J. of Universal Computer Science, Vol. 14, pp. 2110--2157 (2008)
10. Kakousis, K., Paspallis, N., Papadopoulos, G. A.: A survey of software adaptation in mobile and ubiquitous computing. In: J. of Enterprise Information Systems, Vol. 4, pp. 355--389 (2010)
11. Huang, Z., Li, W., Hui, P., Peylo, C.: CloudRidAR: A Cloud-based Architecture for Mobile Augmented Reality. In: Proceedings of MARS '14, pp. 29-34 (2014)
12. Zhang, X., Jeong, S., Kunjithapatham A., Gibbs, S.: Towards an elastic application model for augmenting computing capabilities of mobile platforms. In: Proceedings of 3rd International ICST Conference on Mobile Wireless Middleware, Operating Systems, and Applications (Mobile-Ware), pp. 161--174 (2010)
13. Ionescu, A.: Resource Management in Mobile Cloud Computing. Informatica Economica, Vol. 19, pp. 55--66 (2015)
14. Camara, J., Salaun, G., Canal, C.: On run-time behavioral adaptation in context-aware systems. In Proceedings of M-ADAPT'07 at ECOOP '07. pp. 26—34 (2007)
15. Vekshyn O., Tkachuk M. Algorithmic Software Adaptation Approach in Mobile Augmented Reality Systems. In: Proceedings of 7-th International Conference on Software Engineering Advances, Lisbon, Portugal, pp. 40--43 (2012)
16. Official Web-site of QRCode, http://www.qrcode.com
17. Official Web-site of Barcodes & ID Key Standards, http://www.gs1.org/gsmp/kc/barcodes
18. Official Web-site of QR Droid, https://play.google.com/store/apps/details?id=la.droid.qr&hl=ru
19. Sommervile, I. Software Engineering. Addison Wesley (2011)
20. Tian He et.al. Feedback Control-based Dynamic Resource Management in Distributed Real-time Systems. In: J. of Systems and Software, vol. 80, pp. 997–1004 (2007)
21. Ghanbari, H. Model-Based Dynamic Resource Management for Service Oriented Clouds. PhD thesis, York University Toronto, Ontario (2014)
22. Sun, Y. White, J., Eade, S.: A Model-Based System to Automate Cloud Resource Allocation and Optimization. In: J. of Model-Driven Engineering Languages and Systems, Vol. 8767, p 34 (2014)
23. Fensel, D., Kerrigan, M., and Zaremba M.: Implementing Semantic Web Services: The SESA Framework. Springer-Verlag Berlin Heidelberg(2008)

24. Tenório T., Dermeval D., Bittencourt I.: On the use of ontology for dynamic reconfiguring software product line products. In: ICSEA 2014, Proceedings of the 9th International Conference on Software Engineering Advances, pp. 545--550 (2014)
25. Hervas, R., Bravo, J., Fontecha, J., Villarreal, V.: Achieving Adaptive Augmented Reality through Ontological Context-awareness Applied to AAL Scenarios. In: J. of Universal Computer Science, vol. 19 (9), pp. 1334--1349 (2013)
26. Chuan-Jun S., Tzu-Ning Y., Yu-Ming Y.: Ontology-based Mobile Augmented Reality for Personalized U-Campus. In: Proceedings APIEMS 2012, pp.2037--2046 (2012)
27. Hatala, M., Wakkary, R.: Ontology-Based User Modeling in an Augmented Audio Reality System for Museums. In: J. of User Modeling and User-Adapted Interaction, Vol. 5(3–4), pp. 339--380 (2005)
28. Bārzdiņš, J., Bārzdiņš, G., Čerāns, K., Liepiņš, R., Sproģis, A.: OWLGrEd: a UML Style Graphical Notation and Editor for OWL 2. In: Proceedings of 7th International Workshop “OWL: Experience and Directions”, (2010)
29. Ramesh, K., Karunanidhi, P. Literature Survey on Algorithmic and Non-Algorithmic Models for Software Development Effort Estimation. In: Int. J. of Engineering and Computer Science, Vol. 2 (Issue 3), pp. 623--632 (2013)
30. Aliev A., Soft Computing and its Applications. World Scientific (2001)
31. Maiden, N., Sutcliffe, A. Case-based Reasoning in Software Engineering. In: IEE Colloquium on Case-Based Reasoning, Digest No.036, pp. 1-3, (1993)
32. Limthanmaphon, B., Zhang, Y. Web Service Composition with Case-Based Reasoning. In: Proceedings of 14th Australian Database Conference (ADC2003), Vol. 17, (2004)
33. Tkachuk, M., Polkovnikov, S., Bronin, S. Adaptive Control Framework for Software Components: Case-based Reasoning Approach. In: Proceedings of 6th International Workshop on Software Cybernetics (IWSC – 2009), Seattle, USA, pp. 47--56 (2009).
34. Official Web-site of Android platform, https://www.android.com/
35. Official Web-site of Oracle Berkeley DB, http://www.oracle.com/technetwork/database/database-technologies/berkeleydb
36. Official Web-site of OpenCV project, http://www.opencv.org
37. Official Web-site of PHP language, http://www.php.net
38. Official Web-site of Apache project, http://www.apache.org
39. Official Web-site of MySQL project, http://www.mysql.com
40. Official Web-site of MongoDB , http://www.mongodb.org