The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-07995-0_42

Mining in Dynamically Composed Scripted 3D Scenes for Better Access Control – Computational Evaluation

Adam Wójtowicz

Department of Information Technology, Poznań University of Economics, Poland

awojtow@kti.ue.poznan.pl

Abstract. In this paper computational complexity of an approach called Selective Semantic Modeling (SSM) is evaluated. The algorithm of the SSM allows for user privileges "mining" from dynamically composed scripted 3D scenes with respect to semantics of inter-object interactions. The SSM method is based on the concept of semantic operations, which are generated at run-time from the scripted content and are used to construct user privileges.

Keywords: call graph mining, dynamic content composition, user privileges

1 Introduction

An entirely new level of applicability of the three-dimensional techniques is achievable by the development of technologies that enable describing interactive synthetic multimedia content in a way, which is independent of hardware and software [1][2]. It allows for dynamic composition of the content coming from distributed heterogenous platforms, which creates a developing "market" – scenes composed ad hoc in the cloud can be dynamically published by institutions or individuals, and – at the same time – new users can consume this content independently on their client configurations. Educational instititions could take advantage of this model by releasing for publishing (and potentially subsequent republishing) their content in the form of interactive scripted 3D models [3], or build own interactive scenarios using both own and third party building blocks. 3D interactivity itself is indeed attractive and natural medium for the modern digital citizens.

Nevertheless, in recent years it has been observed that the growth of many 3D environment services, including these supporting user-generated content, is frequently slowed down or even stopped. One of the major reasons for this is related to the user and content security [4][5]. In order to participate and contribute, publishers, content creators as well as users need to be sure that their content and other data will not be misused in any way. Also, the problem of usage control in the context of the IPR assurance is the issue [6]. It requires assuring fine-grained control over confidentiality and integrity of the content. One of the significant elements of the protection mechanism is control of runtime calls of the scripted content that is dynamically composed and run on the remote 3D engine hosts.

A method called Selective Semantic Modeling (SSM) has been proposed by the author in [7][8][9] to address this problem. To protect behavioral content in effective yet unobtrusive and flexible way, it uses privileges "mined" from interactions between scripted objects of persistently running environment. Possible interactions (calls) are analyzed by taking into account the call range of object methods (in scripting languages also functions or procedures). The approach automatically encompasses newly created objects and follows the evolution of the call graph. Privileges are manageable and understandable by both non-human and human operator.

In this paper computational complexity of this new approach that can be used to protect interaction within scripted scenarios where untrusted content is used in the process of dynamic scene composition is analyzed.

2 Related work

2.1 Dynamic 3D content composition

In the context of dynamic 3D content composition, X3D data format devised by the Web3D Consortium must be considered. It permits composing complex 3D scenes from distributed components in two ways, through linking resources and using prototypes. A few projects have focused on building 3D scenes from independent components. The work [10] addresses composition of 3D scenes using a framework built upon web services without extending the syntax of any available 3D content descriptions. Extensions of the X3D syntax have been proposed in the work [11] to enable video streaming from distributed sources. Also X3D-based servers has been released, such as BS Collaborate [12] to provide simple collaborative 3D environments.

Decreasing popularity and research effort related to proprietary platforms for collaborative 3D environments such as Second Life is observed. However, recently significant progress can be noticed in open source software communities. OpenSimulator [13] is an open source engine whose infrastructure leverages several communication protocols for message exchange among different sides: clients, servers and external stations, and enables development of similar environments as Second Life. Services based on OpenSimulator such as Hypergrid [14] support dynamic adding of third-party regions on external hosts to the grid. Open Cobalt [15], yet another open source platform for constructing, accessing, and sharing 3D environments supprots hyperlinking virtual spaces using 3D portals to form a large distributed network of interconnected collaboration spaces. It does not require centralized servers and the processing is distributed in a peer-to-peer manner. One of the other open source platforms is Open Wonderland [16] supporting creation of a wide range of interactive and dynamic environments. Contrary to Open Simulator, Open Wonderland is not based on the paradigm of distribution, thus it suffers all the limitations of the centralized system. Another group of solutions include game engines, e.g., Unity [17] or RakNet [18] convenient for distributed simulations based on proprietary solutions.

In the group of standardized network protocols, a few standards can be distinguished that have been designed for distributed interactive simulations, e.g., Distributed Interactive Simulation (DIS) [19], and High Level Architecture (HLA) [20]. DIS

is a protocol specifying the exchange of messages among participants of the simulation that describe locations, velocities, orientations, and several other features of the units. HLA defines an infrastructure incorporating heterogeneous platforms, their interfaces and responsibilities.

2.2 Security of dynamic 3D content composition

There are many techniques that aim at protecting distributed multimedia data. DRM is a group of techniques to control access and usage of digital content, including multimedia data as described in [21]. However, constantly developing multimedia techniques in conjunction with the development of networking techniques challenge the existing DRM systems. In particular, randomly chosen fragments of 3D scenes, containing behavioural objects (including scripting source code), interacting dynamically with each other and created by distributed users cannot be sufficiently protected by current DRM systems.

The distinguished standardization effort in the domain of protecting usage rights of multimedia content is MPEG-21 REL [22], a rule-based access control language developed for expressing rights related to resources under a set of conditions. However, Digital Item representation, which is the base for this model, is not expressive enough to support complex behaviour-rich 3D scenes with content that has to be protected selectively. Alternative languages, like XACML, or ccREL are even more generic.

3D models security in distributed VR systems is a wide topic. In case of systems based on OpenSimulator [13][14] engine, where new regions on new hosts can be dynamically added to the grid, roaming data processing model and external openness make them inherently insecure, which affects digital items usage control. It cannot be assumed that host software has not been modified in order to make illegal copies of digital items that constitute users inventory. Even more decentralized approach has been applied to Open Cobalt [15], where content processing is distributed in a peer-to-peer manner. Reduction of reliance on error-prone server infrastructures by using a peer-based messaging protocol increases scalability of the approach, but in turn the problem of untrusted client impacting data security appears. In Open Wonderland [16] any object within an environment can be associated with an access control list to control which users can view, manipulate or edit the object. However, a list of possible operations is predefined and does not reflect scenes coded logic. Also X3D-based collaboration servers, such as BS Collaborate [12] provide only limited security measures, which do not enable definition of fine-grained semantically-rich privileges.

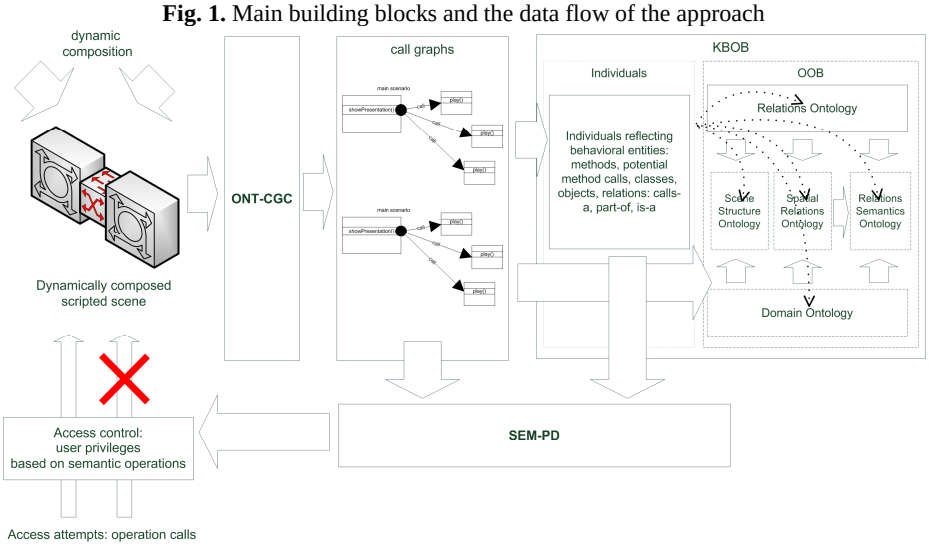
3 Overview of the approach

In a process of composition of distributed scripted content, objects and their behavior encoded in scripting programming languages is created in unsupervised and decentralized manner by users. The structure of the content that is a subject to dynamic composition evolves. To address the problem of controlling the usage of a large number of dynamic and interactive objects in a way that would be both flexible enough to

encompass a large variety of possible operations and understandable and manageable for users, the SSM approach has been proposed by the author [7][8][9].

The concept of the semantic operation aggregating corresponding call graphs has been introduced. It enables the use of operations that are applicable on a higher abstraction level than the level of user-generated methods. At the same time it preserves the encapsulation of the code: it groups existing methods without mixing their code or creating any new methods. Semantic operation is defined as a set of methods with identical call range. Semantic operations reflect all the method calls, i.e., interactions among the objects, which may be dynamic, parameterized and conditional. A privilege is defined as a pair semantic operation – object (following the standard RBAC model, both privileges and users are assigned to roles). The privilege assigned to a role enables a user playing that role to launch an object method, if the method is an element of the semantic operation's set of methods.

Fig. 1. Main building blocks and the data flow of the approach



Conceptually, a semantic operation adds a new dimension to the concept of operation known from standard access control models. Similarly to operation, it is used to define privileges in conjunction with objects. However, they can represent arbitrary methods, e.g., implemented in objects as a part of user-generated content. It is assured that semantic operations are always consistent with methods implementation. Since dependencies of inter-object interactions are not known a priori, method call ranges are automatically analyzed based on the structure of the composed scene.

The privileges are expressive (based on methods, not on predefined operations) and at the same time stable (not dependent on insignificant data changes) and consistent with the evolving content. They are especially useful in large compositions, i.e., having a large number of classes, objects, methods, privileges or roles. If the set of methods forms a common semantic operation, it is clear for privilege managers what the intended purpose of these methods is. The fact that a semantic operation groups all methods with common call range and knowledge of its place in the semantic opera-

tions hierarchy reduces the risk of granting unintended privileges. Unintended privileges could enable an unauthorized user to launch a method that calls other methods, which spreads method calls on methods that should not be called by this user.

The semantic aspect of the approach enables modeling of privileges for behavioral content with respect to the semantics of the behaviors. Call graphs are enhanced (labeled) by using semantic descriptions represented as assignments of individuals, i.e., facts (representing pieces of source code) to ontology classes (representing semantics). These descriptions are both induced from the source code and added as a result of semantic description. To this end, the *Knowledgebase of Objects Behavior* (KBOB) – built according to *Ontology of Objects Behavior* (OOB) – is used (**Fig. 1**). The OOB ontology defines classes and properties for general use in 3D applications as well as those reflecting specificity of a given application.

Classification of individuals representing methods plays the role of the graph node labeling. In the process of semantic unification, to semantically label a call graph represented by the KBOB, knowledgebase querying and reasoning is applied. Due to the application of the OOB ontology formalism, the process of labeling (tagging) methods, classes and objects with descriptions (constructing knowledgebase) results in a consistent semantic description, which can be used to produce more accurate semantic operations than metadata sets or pure call graphs. Human-readable descriptions of semantic operations can be generated, based on call graph labeling.

The OOB is composed of five sub-ontologies related to relations, relation semantics, scene structure, spatial semantics and domain-specific concepts. The sub-ontologies can be applied simultaneously with mutual references, or selectively, according to knowledge that is or potentially can be collected in the knowledgebase. KBOB individuals denote classes, methods, objects, potential method calls, and relations extracted from the source code. These individuals are classified to OOB classes using manual assignments, or preferably, using asserted conditions that are defined or adjusted by trusted ontology contributors. The membership in the OOB classes determines semantics of an individual. Multiple inheritance mechanism is used, so complex semantic descriptions can be created just by using orthogonal class hierarchies and inheritance mechanism.

Both the OOB ontology and the KBOB knowledgebase are subjects to modifications during run-time. Reorganization of the KBOB that is caused by the OOB changes consists in reclassification of individuals, which is equivalent to changing labeling of the call graph. During the scene lifecycle, the OOB ontology is extended by trusted OOB contributors by: creating new subclasses of classes; creating new sub-properties for classes; defining “necessary and sufficient” asserted conditions for classes that enable dynamical classification of individuals to classes according to some custom conditions. The process of construction of the KBOB knowledgebase begins with creating a piece of behavioral content (source code). Then individuals (methods, potential method calls, classes, objects) are automatically added to the KBOB by source code analysis algorithm ONT-CGC, and they are are automatically classified based on existing OOB classes definitions (this plays the role of semantic tagging). Some specific individuals are manually classified. In the process of semantic unification, the proposed SEM-PD algorithm is used.

4 Algorithms evaluation

A call graph is a basic data structure used to generate semantic operations by analysis of method's call subgraphs mutual correspondence. The approach follows the evolution of call graphs by regenerating the semantic operation set on each change. The semantic operation induction is based on a static call graph – a graph whose nodes are methods and edges are all possible calls of other methods. A call graph is a directed graph. In the source code, each method is defined once in its class, but in the call graph each method is represented by separate nodes for all the objects being instances of the class. If a given class has many objects as instances, then for each method of this class there are as many nodes as objects. Edges represent method calls from one method to another. The order of the calls obtained from the source code is preserved: the call graph is an edge ordered graph, which means that ordering of the outgoing edges is stored for each node in the form of a list. It is worth noting that, in general, a call graph is not a tree, because two different methods can call a common method, and even it is not a DAG because it can contain cycles.

The call graph is both node-labeled and edge-labeled. Each node is labeled with identifiers of a class and a method. Edge labels represent parameters that are passed to the called method. If values of these parameters are known – i.e., in a case when they do not depend on run-time user input – parameter values can be reflected in the labels as well. In such a case, each edge label is a list of pairs, each pair is composed of a parameter identifier and its value. Global and local call graphs are distinguished. The global call graph contains all the methods and all the potential calls of other methods included in the source code of all the objects at a given moment. It is updated incrementally when the source code changes. A local call graph is a subgraph of the global graph built starting from a given method and containing all the methods that are called by this method and all their callees. Despite the fact of the regeneration of the call graph at run-time, it is not a dynamic call graph, i.e., it is not a result of the analysis of actual run-time calls, but the result of analysis of the method call dependencies found in the code at run-time.

4.1 Analysis of ONT-CGC algorithm

The first main algorithm used is ONT-CGC algorithm for construction of the global call graph. The construction is performed in two modes: the *initial mode* and the *incremental mode*. The initial mode is used when the global call graph has to be created from scratch before the first application, or after its major remodeling. The incremental mode is used after each change of the scripted objects concerning method calls. The result of the algorithm executed in the incremental mode is an update of the global call graph, which usually concerns its small part only.

All script commands (their number is denoted by N) are analyzed in order to add methods (nodes) and arcs (calls) to the call graph. For each method there are nodes created – as many nodes as many objects there is in the code, the number of objects is limited by N. Therefore computational complexity is $O(N^2)$. For each command (N)

the algorithm stores the state of every variable, and the number of variables is limited by N, thus space complexity of the algorithm is $O(N^2)$ as well.

4.2 Analysis of SEM-PD algorithm

The aim of SEM-PD is to compare subgraphs by verifying an isomorphism preserving labeling between call graphs. The set of semantic unification criteria is a parameter of the SEM-PD algorithm. Each semantic unification criterion is represented as an ontology class. In the semantic unification process, all sub-classes of each class specified as a unification criterion are analyzed.

The algorithm finds all identical subgraphs of the global call graph, i.e., local call graphs. The key point is definition of conditions under which two local call graphs are considered to be identical. The approach provides three alternative variants of the algorithm that can be selectively chosen according to application requirements. In all these variants, two local call graphs are identical if there is an isomorphism between those two graphs, which preserves labels.

In the SEM-PD algorithm, each method can be potentially a starting point for building a local call graph and each pair of methods can be potentially a subject of semantic unification. However, for performance optimization purposes, not each pair of methods is in fact compared: local call graphs to be compared are first pre-selected. Pre-selection uses subgraphs measures (size, diameter, etc.) to filter subgraphs candidates that are likely to match. This mechanism is required after each global call graph update at runtime. It does not introduce a risk of omitting two identical local call graphs comparison since what it does is just skipping the comparisons of obviously different local call graphs.

Table 1. Analysis of computational complexity of SEM-PD elements;
n – number of nodes in global call graph;
k – number of nodes in local call graph corresponding to added method (k<<n)

Phase	**Operation**	**Algorithm step**	**Typical complexity**	**Rare worst-case complexity**
Initial				
	Subgraphs hashing	Calculation of all hash values	**$O(n^3)$**	
	Subgraphs matching	Comparison of all hash values	**$O(n^2)$**	For comparing equal hash values – isomorphism preserving labeling is verified. Complexity linear for a pair, - generally $O(n^3)$, if not trees O(n!).
Incremental				
	Adding/ modyfing a node	In case of different hash values: - finding new value; - comparing it with all others.	**$O(k^3+n)$**	
	Adding/ modyfing a node	In case of equal hash values: - finding new value; - comparing it with all others; - isomorphism test.	**$O(k^3+n^2)$** the case of tree	$O(n \cdot k!)$ the case of unrestricted graphs; k<<n

In **Table 1** typical and rare worst-case computational complexity of SEM-PD algorithm is listed. Since domain of application the is security critical access control, no heuristic approach producing approximate results has been applied. In rare case of equal hash values exact verification is performed, and in case of different hash values, these values unambigously confirm lack of isomorphism. It is worth to note that for graph isomorphism problem that applies to one of the usage case, for general unrestricted graphs (as well as for labeled graphs) no polynomian-time algorithm has been proposed in the literature (but also it is not proved that it belongs to the NP-c class). Polynomian-time algorithms are proposed for trees (and are used in SEM-PD algorithm), and for other specific graphs, e.g. planar graphs (here planarity cannot be assumed).

Practical usefullness of the presented approach during the runtime is a fact, despite expotential complexity in rare worst case, due to:

- Using hash values, equal for identical graphs, which makes graph comparison fast
- Using incremental approach, limiting analysis to local call graphs
- Nature of call graphs generated from scripting code: global call graphs are sparse graphs, containing many isolated subgraphs, while local call graphs are frequently trees; there is large number of them, and they have low number of nodes;
- Operations of creating and modyfying privileges, as well as using privileges (executing methods), which are the most frequent operations taking place during environment runtime, are performed without additional computational overhead dependent on problem instance.

5 Discussion

Application of the proposed approach provides three factors reducing barriers of mass adoption of interactive 3D content that is dynamically composed. First, user rights to the content are protected, which encourages users to contribute. Second, user participation can be deeply interactive (users can create objects that interact with other users' objects), which makes content more attractive. Finally, synergy effect can be obtained since not only content but also its security attributes originating from different distributed sub-systems can be seamlessly integrated (contrarily to both traditional low-level privileges and resources and high-level roles coming from heterogenous systems with hardly interpretable semantics).

Distributed nature of dynamic content composition intensifies the need of automation and machine-aided management of user privileges and their semantics across content sources. With the presented approach middleware aggregating distributed scenarios is responsible not only for bridging data and events, and synchronization tasks, but also for security interoperability and management of access control. Most of the approach functionality: call graph similarity analysis, semantic operation induction and notification, knowledgebase maintenance can be realized in the middleware designed for dynamic composition of the content. Semantic consistency of privileges in dynamically composed scenes is forced even if they are based on different technologies and data models.

Using evaluated approach it is possible to effectively automate not only the induction of semantic privileges but also the detection of an attack (malicious calls) or a security hole (potential method call combination), which can be useful especially when used in conjunction with dedicated or general purpose Intrusion Detection System, such as proposed in [23][24]. Given calls can be compared with known dangerous call patterns (the simplest pattern: data harvesting) by the call graph matching, which is originally used in the approach to induce semantic operations. For this purpose artificial "malicious" semantic operations may be constructed and compared with existing call graphs. Other data security threats, regarding user-generated in-world source code, such as automation attacks, risks to intellectual property, spam or attempts of denial of service can be detected as well. Also, injection of code calling unintended methods can be detected. Once an attack on some objects or the presence of a security hole is detected, it can be neutralized, since the identification of operation corresponding to dangerous method call graphs and global deactivation of all privileges that use it is fast and in-depth.

6 Conclusions

There is a significant obstacle for making 3D scripted scenes, both user-generated and provided by the institutions, available to the dynamic content composition. This is the problem of the user and data security**.** If a user cannot be sure whether they are able to control the access of other users to their content, they will not participate actively in the content development. Moreover, if they cannot be sure whether their interactions with the content created by other users is always safe for them, their privacy and their content, probably they will not explore the environment at all.

In the approach presented in this paper, security mechanism based on the concept of semantic operations is evaluated. The evaluation proves theoretically that the algorithm proposed in the presented approach has the feature of scalability which is required for dynamic composition of large amounts of content. From the practical point of view, it is important that proposed soultion is interoperable with standard access control models as well as with interactive scenarios data standards, constituting a middle layer, and designed with respect to the specificity of 3D scenarios semantics.

Acknowledgements. This research work has been partially supported by the Polish National Science Centre Grant No. DEC-2012/07/B/ST6/01523.

7 References

1. Eno, J., Gauch, S., Thompson, C.: Searching for the metaverse. In: Proc. of the 16th ACM Symp. on Virtual Reality Software and Technology, 223-226, ACM, New York, (2009)
2. Botev, J., Hohfeld, A., Schloss, H., Scholtes, I., Sturm, P., Esch, M.: The HyperVerse: concepts for a federated and Torrent-based '3D Web'. In: Int. J. Adv. Media Commun. 2, 4, 331-350 (2008).

3. Flotyński, J., Dalkowski, J., Walczak, K.: Building multi-platform 3D virtual museum exhibitions with Flex-VR, In: The 18th International Conference on Virtual Systems and Multimedia, 391-398, IEEE Advancing Technology for Humanity, Milan (2012)
4. Horn, D., Cheslack-Postava, E., Azim, T., Freedman, M.J., Levis, P.: Scaling Virtual Worlds with a Physical Metaphor. IEEE Pervasive Computing 8, 3, 50-54 (2009).
5. Alpcan, T., Bauckhage, C., Kotsovinos, E.: Towards 3D Internet: Why, What, and How?. In: Proc. of the 2007 Int. Conf. on Cyberworlds., 95-99, IEEE, Washington, DC, (2007)
6. Lian, S., Kannelopoulos, D., Ruffo, G.: Recent Advances in Multimedia Information System Security. Informatica, 33, 1, 3-24 (2009)
7. Wójtowicz, A.: Secure User-Contributed 3D Virtual Environments, in: Interactive 3D Multimedia Content – Models for Creation, Management, Search and Presentation, Springer, London, 171-193, Springer, London (2012)
8. Wójtowicz, A., Cellary, W.: Representing User Privileges in Object-Oriented Virtual Reality Systems. In: Camarinha-Matos L., Pereira P, Ribeiro L. (eds.) Emerging Trends in Technological Innovation, IFIP series, 52-61, Springer (2010)
9. Wójtowicz, A., Cellary, W.: Access Control Model for Dynamic VR Applications. In: Rea A. (ed.) Security in Virtual Worlds, 3D Webs, and Immersive Environments: Models for Development, Interaction, and Management, 284-305, IGI Global (2011)
10. Zhang, X., Gracanin, D., Service-Oriented-Architecture based Framework for Multi-User Virtual Environments. In: Proceedings of the 40th Conference on Winter Simulation. Miami, 1139-1147, USA, (2008)
11. Repplinger, M., Löffler, A., Schug B., Slusallek, P.,. Extending X3D for distributed multimedia processing and control. In: Proc. of the 14th Int. Conf. on 3D Web Techn. Darmstadt, 61-69, ACM, New York, (2009)
12. BS Collaborate http://www.bitmanagement.com/products/server/bs-collaborate.
13. Open Simulator http://opensimulator.org/wiki/Main_Page.
14. Hypergrid, http://opensimulator.org/wiki/Hypergrid
15. Open Cobalt http://www.opencobalt.org/.
16. Open Wonderland http://openwonderland.org/.
17. Unity, http://unity3d.com/
18. RakNet, http://www.jenkinssoftware.com/
19. IEEE Standard for Distributed Interactive Simulation—Application Protocols, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=729408
20. IEEE Standard for Modelling and Simulation (M&S) High Level Architecture (HLA), http://simlab.gyte.edu.tr/docs/
21. Zeng, W., Yu, H., Lin, C.: Multimedia Security Technologies for Digital Rights Management. Academic Press, Inc., Orlando (2006)
22. Wang, X., Demartini, T., Wragg, B., Paramasivam, M., Barlas, C.: The mpeg-21 rights expression language and rights data dictionary. Multimedia, IEEE Transactions on 7, 3, 408-417 (2005)
23. Herrero, A., Zurutuza, U., Corchado, E.: A neural-visualization ids for honeynet data. International journal of neural systems 22 (02)
24. Herrero, A., Navarro, M., Corchado, E., Julián, V.: RT-MOVICAB-IDS: Addressing real-time intrusion detection. Future Generation Comp. Syst. 29(1): 250-261 (2013)