[Gupta, Stafford, 2006] J.N.D. Gupta, and E.F. Stafford Jr. Flowshop scheduling research after five decades. European Journal of Operational Research. 169, 699-711, 2006.

[Jackson, 1956] J.R. Jackson. An extension of Johnson's results on job lot scheduling. Naval Research Logistic Quarterly, 3, 201-203, 1956.

[Jansen, Mastrolilli, Solis-Oba, 2005] K. Jansen, M. Mastrolilli, R. Solis-Oba. Approximation schemes for job shop scheduling problems with controllable processing times. European Journal of Operational Research. 167, 297-319, 2005.

[Johnson, 1954] S.M. Johnson. Optimal two- and three-stage production schedules with setup times included. Naval Research Logistics Quarterly, 1, 61—68, 1954.

[Ku, Niu, 1986] P.S. Ku, and S.C. Niu. On Johnson's two-machine flow-shop with random processing times. Operations Research. 34, 130-136, 1986.

[Leshchenko, Sotskov, 2005] N.M. Leshchenko, and Yu.N. Sotskov. Two-machine minimum-length shop-scheduling problems with uncertain processing times. In: Proceedings of XI International Conference "Knowledge-Dialogue-Solution", June 20-24, Varna, Bulgaria, 375-381, 2005.

[Pinedo, 1995] M. Pinedo. Scheduling: Theory, Algorithms, and Systems. Prentice-Hall, Englewood Cliffs. 1995.

[Slowinski, Hapke, 1999] R. Slowinski, and M. Hapke. Scheduling Under Fuzziness. Physical-Verlag, Heidelberg, New York. 1999.

[Shafransky, 2005] Ya.M. Shafransky. Scheduling problems with uncertain parameters: research directions and some results. Informatika. 3, 5-15, 2005 (in Russian).

Authors' Information

Natalja M. Leshchenko – United Institute of Informatics Problems of National Academy of Sciences of Belarus, Surganova str., 6, 220012, Minsk, Belarus; e-mail: leshchenko@newman.bas-net.by

Yuri N. Sotskov – Prof., DSc. United Institute of Informatics Problems of National Academy of Sciences of Belarus, Surganova str., 6, 220012, Minsk, Belarus; e-mail: sotskov@newman.bas-net.by

ACCESS RIGHTS INHERITANCE IN INFORMATION SYSTEMS CONTROLLED BY METADATA

Mariya Chichagova, Ludmila Lyadova

Abstract: All information systems have to be protected. As the number of information objects and the number of users increase the task of information system's protection becomes more difficult. One of the most difficult problems is access rights assignment. This paper describes the graph model of access rights inheritance. This model takes into account relations and dependences between different objects and between different users. The model can be implemented in the information systems controlled by the metadata, describing information objects and connections between them, such as the systems based on CASE-technology METAS.

Keywords: access control mechanisms, graph model, metadata, CASE-technology.

ACM Classification Keywords: D.2 Software engineering: D.2.0 General - Protection mechanisms.

Introduction

As information systems become larger and more complex, and as the number of their users increase, there are growing needs for methods that can simplify and even partly automate the process of access rights assignment.

The main problem of traditional access control mechanisms is that they don't take into account the relations between information objects.

The technology METAS [Lyadova, 2003], [Ryzhkov, 2002] allows to create dynamically adjusted information systems. Means of adaptation are based on use of the metadata describing information objects and connections between them from the various points of view. Functioning of information system is carried out through interpretation metadata by MDK METAS The metadata can form a basis for realization of mechanisms of the rights equivalence, in particular, the mechanism of the access rights inheritance that allows to lower labour input of assignment of the rights the manager of system. At the same time there are problems at definition of the rights of access on the objects accessible on several relations from parental objects to which various rights are appointed.

The model proposed in this paper take such relations in consideration and the rules for it are formulated.

To define the inheritance mechanism we shall formally describe model of distribution of the access rights. As basic elements of the model are used access graph and rules of its transformation.

Graph Model

An information system consists of objects and subjects. Access control describes whether specific subject can access specific object.

Let O is a set of objects and S is a set of subjects. $G = (V, E)$ is a finite directed labelled graph, where $V = O \cup S$ is a set of *nodes* and E is a set of *arcs*.

Notation $v_i \rightarrow v_j$ means that there is an arc $(v_i, v_j) \in E$ in graph G. A node $s_i \in S \subseteq V$ is called *subject-node* and a node $o_i \in O \subseteq V$ is called *object-node*.

If $v_i \rightarrow v_j$ and both of these nodes are subject-nodes (or both of them are object-nodes) node v_i is called *parent* of node v_j and node v_j is called *child* of node v_i.

Subject-nodes which have not got any children are called *users* all other subject-nodes are called *groups*.

Object-nodes which have not got any children are called *leaves* all other subject-nodes are called *roots*.

Arcs between objects present the relations between them. The arc's direction depends on type of relationship between objects:

- 1 : 0..1 $\Rightarrow$ arc from "0..1" to "1";
- 1 : M $\Rightarrow$ arc from "1" to "M";
- 0 : 1..M $\Rightarrow$ arc from "0" to "1..M";
- M : M $\Rightarrow$ bidirectional arc.

For example, for part of the database scheme which is shown on fig. 1 the subgraph on fig. 2 is fitted.

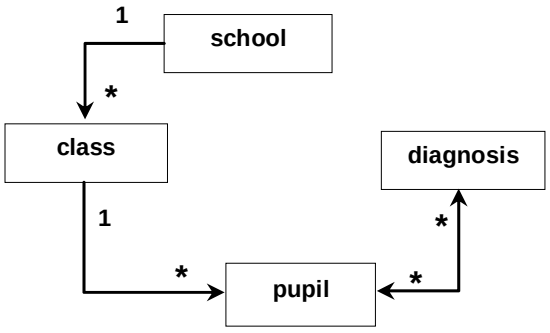

Figure 1. Database Scheme Fragment

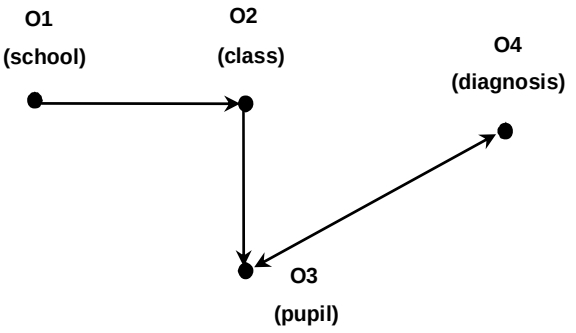

Figure 2. Object-nodes Subgraph

Each of the subjects must be connected by the arc to each of the objects. An arc between subject-node and object-node is called *access arc*.

An access arc's label determines the assigned access right of this subject to the object. *Assigned access right* is determined by the information system's administrator and it can deny or allow access to information objects.

Access rights that take into account relations between subjects and objects are called *actual access rights*.

Subject-nodes

In this section relations between subjects are considered and the rules that can take them into account are formulated. Access rights that allow for relations only between subjects are called *actual subject's access rights*.

Let $parent(s_i) = \{s_j \in S \mid \exists (s_j, s_i) \in E\}$ – is a set of parents for subject-node $s_i \in S$.

Let access arcs have following labels:

- $right(s_i, o_j) \in \{0, 1, 2, 3\}$ is an *assigned access right* of subject $s_i \in S$ to object $o_j \in O$, where 0 means that right is not assigned, 1 – access is denied, 2 – subject has a partial access and 3 – access is allowed. *Partial access* means that access is allowed only if certain conditions are fulfilled. These conditions are determined by administrator.
- $a\_right(s_i, o_j) \in \{1, 2, 3\}$ is an *actual subject's access right* of subject $s_i \in S$ to object $o_j \in O$, where 1 means that access is denied, 2 – subject has a partial access and 3 – access is allowed.

The subject's access rights can depend on its parents' rights. The process of determination the actual subject's rights is called *subject's rights inheritance*.

Let $s_i \in S$. The inheritance should be done according to the following two rules.

Rule 1. If subject has an assigned right $right(s_i, o_j) \neq 0$ to object $o_j \in O$, the actual subject's right is determined as $right(s_i, o_j)$, i.e.

$$a\_right(s_i, o_j) = right(s_i, o_j) \tag{1}$$

The main idea of this rule is that explicit assignment is more significant than inheritance.

Rule 2. If an access right to object $o_j \in O$ isn't assigned, i.e. $right(s_i, o_j) = 0$, the actual subject's right is determined as maximum of its parents' actual rights :

$$a\_right(s_i, o_j) = \max_{s_k \in parent(s_i)} (a\_right(s_k, o_j)) \tag{2}$$

If an access right to object $o_j \in O$ isn't assigned and the subject hasn't got any parents its actual right is determined as 1. It means that access is denied.

Note that by definition the maximum value for *a_right* is 3 (that means that access is allowed).

For finding actual subject's rights the above two rules are recursively applied.

Object-nodes

In this section relations between objects are considered and the rules that can take them into account are formulated. Access rights that allow for relations only between subjects are called *actual object's access rights*.

Note that the same object can be connected with different objects and rights can depend on the object from which the access is done.

As access rights depend on access context let define $context(s_i, o_j)$ as a current *access context*, i.e. list of object-nodes (path from one of the roots to current object-node).

Parent from context is an object-node from the access context which is the parent for node $o_j \in O$. Let $c\_parent(o_j)$ is a parent for object-node $o_j \in O$ from context.

Let access arcs also have following labels:

- $o\_right(s_i, o_j) \in \{1, 2, 3\}$ is an *actual object's access right* of subject $s_i \in S$ to object $o_j \in O$, where 1 means that access is denied, 2 – subject has a partial access and 3 – access is allowed.

Let arcs between object-nodes have the following labels:

- *inherit*(o_k, o_j) ∈ {*true*, *false*} shows the possibility of inheritance. Let *inherit*(∅, o_j) = *false* that means that inheritance from empty object is forbidden.

The process of determination the actual object's rights is called *object's rights inheritance*.

Let $s_i \in S$. The inheritance should be done according to the following three rules.

Rule 3. If subject has an assigned right *right*(s_i, o_j) ≠ 0 to object $o_j \in O$, the actual object's right is determined as *right*(s_i, o_j), i.e.

$$o\_right(s_i, o_j) = right(s_i, o_j) \quad (3)$$

This rule is the same as the rule 1 for subjects' rights inheritance.

Rule 4. If an access right to object $o_j \in O$ isn't assigned, i.e. *right*(s_i, o_j) = 0, and *inherit*(o_k, o_j) = *false* where o_k = *c_parent*(o_j) the actual object's right is determined as prohibition of access, i.e.

$$o\_right(s_i, o_j) = 1 \quad (4)$$

This rule means that if the inheritance is forbidden in current context and there are no assigned rights the access is denied.

Rule 5. If an access right to object $o_j \in O$ isn't assigned, i.e. *right*(s_i, o_j)=0, and *inherit*(o_k, o_j)=*true* where o_k=*c_parent*(o_j) the actual object's right is determined as follows:

$$o\_right(s_i, o_j) = o\_rigth(s_i, c\_parent(o_j)) \quad (5)$$

In order to determine actual access rights in rules 3, 4, 5 we should use *a_right* instead *right*.

Conclusion

Using of access rights inheritance allows to simplify the access rights assignment by automatic taking into account relations between object and subject. This method also permits to avoid some kind of mistakes which can be made by information system's administrator.

In addition to the means described in the given article means of the control of correctness of obvious assignment of the access rights for objects and their attributes are offered also [Mikov, 2003].

The architecture of the CASE-system METAS, the models of the metadata used in this system, and principles of its functioning are described in several articles [Lyadova, 2003], [Ryzhkov, 2002].

The system METAS is developed on the .NET platform. The technology ADO.NET, providing independence from DBCS, is used for access to the objects in database.

Bibliography

[Lyadova, 2003] L.N. Lyadova, S.A. Ryzhkov. CASE-technology METAS. In: Scientific Articles Collection "Mathematics of Program Systems". Perm University, Russia, 2003, pp. 4-18.

[Mikov, 2003] A.I. Mikov, M.V. Chichagova. The Control over Rights Assignment. In: Scientific Articles Collection "Mathematics of Program Systems". Perm University, Russia, 2003, pp. 207-212.

[Ryzhkov, 2002] S.A. Ryzhkov. The Concept of the Metadata in the Development of Information Systems. In: Scientific Articles Collection "Mathematics of Program Systems". Perm University, Russia, 2002, pp. 25-35.

Authors' Information

Mariya Chichagova – Perm State University, Assistant of the Department of Computer Science; PSU, 15, Bukirev St., Perm, 614990, Russia; e-mail: chichagova@dom.raid.ru

Ludmila Lyadova - Institute of Computing, Deputy Director; 19/2-38, Podlesnaya St., Perm, 614097, Russia; e-mail: lnlyadova@mail.ru