Volume 3, Issue 7, July 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Empathy of Access Controlcharacteristics of Heterogeneous Software Components in Uniframe Framework

Dhowmya Bhatt
Research Scholar(CSE)
Mewar University, Chittorgarh, India

Ekata Gupta
Associate Professor, Department of Mathematics
Krishna Institute of Engineering and Technology, India

Abstract- The convention of wireless Networks have almost become inevitable for the human community. The operations also have shifted to distributed computing (DCS), parallel Computing or grid computing from being centralized depending on the implementation necessities. One advantage of distributed computing is its flexibility to adapt to code reusability which is the most important feature of Object Oriented Programming. Extending the concepts of Object Oriented Programming, Component based Software Development has made theseobjects that are self-explained in certain environments and accessed to have a public interface in a uniframe with private implementation.Uniframe is one such approach that facilitates the interaction of heterogeneous software components. Through uniframe DCS can be fashioned to work in a semi-automated facilitating the entire working to be transparent and characteristics predictable. Now its also equally essential to develop a mechanism that will describe, analyse and research the security characteristics of system. Then characteristics can be decomposed into properties that can search software componentsseparately and then combined as an integrated system which then will be capable of producing multi- layer security in larger industry. This research paper focuses on one security component "Access Control" that implements guards filtering information access by the user (as contained resource) and denial to unauthorized usage (as a protected resource). This model driven access control will attempt identifying the protected components and also will explain thehow the components guarded by access control guards are identified from the other unprotected components and individual component implementations of both protected and the unprotected components that will be integrated together as a system.An algorithm proposed in this research accounts for searching of the guarded components. The implementations, advantages and scope of the proposed method along with the future directions and study are indicated towards the end of the research paper.

Keywords – access guards,UMM, contained resource, protected resource, Matching Algorithmic Procedure, PROLOG

1. Introduction

The Uniframe is an approach in computing that provides for a complete or "one-piece" interaction of heterogeneous of software components. This helps to minimize major changes in code when this frame is used in different environments with no compromises in the strength of security provided.[5] These components can exist independent of each other in an organization and for research study upon an organization where this frame with access control is implemented; there are two types of changes possibly foreseen.

- If the current employee or the so termed user quits the firm, then the kind of precautionary measures to be introduced to safeguard the confidentiality of the information though not altering too much the currently used access control schema.

It has to be noted that merely changing the password is not the solution as we are dealing with a huge organization consisting of more than a thousand employees. So an alternate from ordinary move will be the only long term security measure.Major changes in source code to increase the strength of security are not also permitted as accordance the DCS uniframe implementations stated. The second case,

- If the access control schema is duped by a third party, then what countermeasures should be implemented to run the entire system back on track with the same conditions stated for DCS Uniframe. [9]

The components are capable of existing independently but can be combined with ease to work to achieve higher performance. The basic aim is to accomplish the highest efficient performance level after all the components are integrated as one system. The Components needed to be developed are detected through the Headhunters and two or more components are intervened by Internet Component Broker by Adatpter technology.[14] Wrap and Glue techniques are used to attain interoperability when put in any of the computing environment.

The Uniframe infrastructure focuses on two major aspects with basic reference to Component Based Software Development,

- Component Development as per system requirement and Deployment of quality tested and checked components.

- System Integration achieved through the amalgamation of deployed components.

The components developed, deployed and integrated are then put into a frame to perform a specific task. Security can be implemented in two levels (i) in a whole-some basis (ii) at certain levels. The access control properties of certain software components are extricated from the main frame of the system as to avoid major changes in coding when the same components are used in that particular uniframe for applying security.[6]

2. Access Control In Uniframe

The best possible solution that can provide efficient access control solution in a huge organization is to design a model that can be platform independent. Through this we can also elaborately characterize access control properties of individual components with the Meta-Modelspecifications for these components. Then these features can be combined to framework of Unified Meta Model that can focus on more on access control properties of each component that will enhance the strength of safety.[2] This is also facilitated by the model Driven Architecture developed by Object Management Group that transforms platform independent models to platform specific models that can contain the Access Control Points where user priorities shall be defined and the nature of access within and outside the organization will be specified.[9]

In order to define access control properties in a distributed environment, it is essential to know and analyse the following parameters of that system

- the resource that has be to access controlled
- the points at which access control has to be implemented.
- the application specific information required where access control is established.

Now the next step will be model these components with access control properties so that a layer of security will be created and implemented for all or certain components as per the need.

In accordance to the proposed model driven access control, the system should be able to handle successfully two different cases.

- Access information without any modifications
- Access denied and login failure reported for further investigations.

Further the model also supports two basic essential functionalities of creating new details of employees and saving the already existing records after any modifications are done by authorized source.

3. System Architecture Of The Proposed Model

The uniframe architecture for supporting the model driven access control is designed to have two general components to be administered by organizational side and three components where the employee access is granted with the priority levels predefined. The attempt will be made to retrieve information of the user/employee and the work that is currently being pursued through each on the servers.[13] The various sections included in this architecture are distributed across multiple servers with their functionalities defined. Access control verifications are enabled for various components for each action by the user.

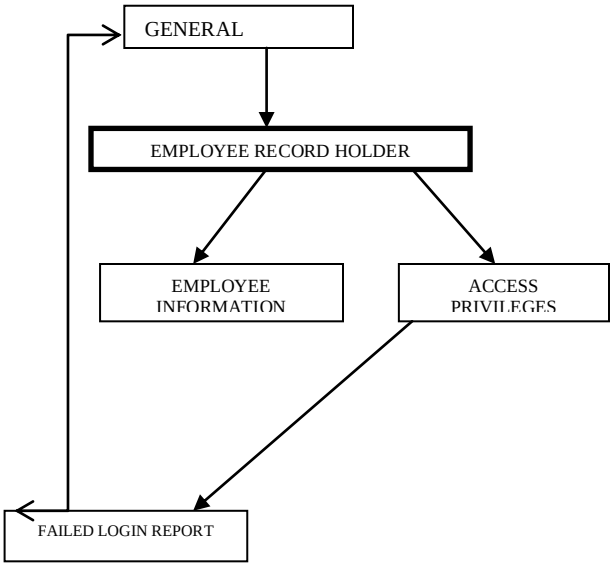

FIG 1 – Architectural design of the model with data flow path and access guards have to be implemented on various layers of each component.

The general or the universal components through which an authorized user can access the data are

- General terminal (GT) – login page; the user information is submitted here and is routed through the employee record holder
- Employee Record Holder(Server) (ERHS) – Information storage

The prioritized components with special access are

- Employee Information (Server) (EIS) – holds transcripts of individual user and his senior employee will be able to access information through this portal.
- Access Privileges (Server)(APS) – the information regarding the current project the employee works upon and the related details according to his level (Developer/ Project-Head/ Project Manager/ COE) will be displayed.

Each of these components supports one or few interfaces integrated together to form the entire system.This information of the sub-components/classes/ abstract types is declared in the Unified Meta Model specifications for each abstract component type in the system. The precise of description of the distribution of these interfaces across the components in the system and how they are specified in Employee Record Holder Server (ERHS) is explained below.

DIEmployeeAdmin – to create employee records whenever a new user joins.

DIEmployee Record – User Accounts, Personal and Professional details of current employees.

Employee Information Server (EIS) is basically designed to hold the professional information of the employee that can be viewed across various destinations of the organization. The information available shall be visible only to the employee and those users who are in a higher level of priority of usage than the employee. Care is taken that the details present in this server are not available for the other employees in the organization than the ones concerned.

DEProfessionalRecord – this interface will contain the current project the employee is working with and past records like experience and achievements.

DEProjectRecord – the current project status of employee can be viewed here. The day to day progress and weekly reports can be generated.

Access Privileges Server(APS) is the main access control gates directly with the user connect. The user can access information in order of priority level in the organization. The COE is the highest level on the priority list and the employee is in the lowest level of access with minimum information admittance.

DMaxpriority – the facility to view information to the users in the highest level of priority.

DintPriority – the Project managers and Technical Heads of various departments are put under this level.

DMinPriority – the employee who is the lowest level of protocol is able to get access of only that information with which directly connected with.

One of these three components shall be the failed login reports about which this research shall not focus much upon because this is entirely a different study that can be elaborated into another research. So as of now only two main components will be taken for the research study with the main the focus upon implementing access control over information access within an organization in an uniframe with application based extrication of characters of software components.

Besides the component interfaces created for the interaction of various components, remote interfaces are also created for those user logins that involves component interfacing. So all of the seven component interfaces that have been implemented will have their own remote interfaces. This will permit operations to be performed through its permitted level of access and prevent unauthorized login.

4. Interaction, Specification And Searching of Components In Proposed Uniframe Model

Now it is equally important to understand how the four major components interact with each other as the system functions sequentially.In each of the component interfaces there are certain specific commands that determine the success or failure of the component interaction and eventually the performance of the model.

While addressing the first and foremost characteristics of access control model are,

- Resource Naming [8]
- Resource identification with access control guards. [15]

Through the process of Resource naming the system Integrator will inform the developer regarding those specific resources that have to be protected by access control guards. Then the resource identification with guards is done through "Matching Algorithmic Procedure" (MAP)

The MAP proposed in this research performs the two basic functions after implementation.

- Identifies those resources within the component that are protected by access control guards from those resources that are unprotected within the same component.
- The system integrator will be able to immediately able to recognize those protected components already existing if certain changes have to be made accordingly in the existing framework.
- This speaks for code reusability. The existing framework need not have to be changed completely if in case there are new resources to be added then only access control guards have to be implemented for newly added resources.
- In accordance to OOP and CBD concept, the model is semi-automated and its functioning will not fail even during the implementation of the new resources.[7] [12]

The MAP for access control properties proceeds as given. The components given by the system Integrator to be protected to the developer is RP. The MAP checks if the resource is inside the component or is an independent. If the resource is found inside the component then check if it is already protected by the component. If the resource sought for is found inside the component and is also protected then the searching is success and the result is YES and if the result is NO then search fails.

In Uniframe Resource Discovery Service (URDS) the matching of the guarded components is done by the Headhunters as mentioned in the introductory part of this research paper. This checks if the access guards implemented by the components are sufficient enough to protect its resources from unauthorized users.[10] Logic Programming (PROLOG) is used instead of other imperative programming languages because PROLOG focuses much on the form of result, the environment and the information which is highly suitable for Uniframe. This is implemented through the predicate calculus where the logical statement states the relationship of the objects. For example, employee(swdeveloper) here swdeveloper is a relation where a dbhatt might be an employee.

5. Implementaion of Access Matching Using Logic Programming In Uniframe

Using the concepts of PROLOG which is one of the most prominent logic programming languages, we aim to protect a particular resource in our component thereby the access control characteristics of the protected components are empathized from those unprotected resources through matching.[11] So the original PROLOG predicates are modified to extract a special matching algorithm developed in this research and called Access Matching Algorithm (AMA).[4] By defining those access control characteristics of resources, we are able to also search those resources that are protected by access guards and which components require access control can also be specified. To protect a resource we use,

require_protect(['*.db',’*.se’’])

the above expression indicates that the employee record '*.db' requires to be protected Extending the same concept to searching the protected the resources contained within a component, the following statements indicate that Access Privilege Server (APS) that contains the complete professional details of a particular employee and can be accessed only by someone who is in senior cadre.

contain('AccessPrivilegeServer ','*.se').

ensure_protect('AccessPrivilegeServer ','*.se').

Consideration is also given to those components that neither contains a resource nor protects one. This is indicated through the predicate, consider(' AccessPrivilegeServer')

The next step is in PROLOG is to match the components according to the specifications and then find whether the resource is protected by component or is in the list of resources that need access guards. For this, initially it is presumed that all components protect a null list of resources. Then PROLOG returns a YES if the resource needs protection predicate is given as, [16]

protects(_,[]).

protects(Component,[Resource|OtherResources]) :-

(contain(Component,Resource),ensure_protect(Component,Resource) ;

A NO is returned if component does not contain that particular resource. This is true also for those components that contain resources that do not need access guards.

(not contain(Component,Resource))),

protects(Component,OtherResources).

match(Component) :- consider(Component),

require_protect(Resources),

protects(Component,Resources).

The above predicates put together will form the base of the Access Matching Algorithm(AMA) that will compare the details in Access Privilege Server(APS) by executing the predicate,

match('AccessPreviligeServer')

6. Empathy of Guarded Resources Using The Proposed Access Matching Algorithm

The Access control characteristics of resources will play an important role when the matching process is done upon the components to determine which of the contained resource is protected and the ones that will need protection. Hence an effective AMA [7] implemented through the PROLOG is proposed to meet out the most important task of matching the components and thereby empathizing those protected resources. The PROLOG predicate is as follows,

require_protect(['*.se']).

consider(' AccessPreviligeServer ').

contain(' AccessPreviligeServer ','*.se').

ensure_protect(' AccessPreviligeServer,'*.se').

protects(_,[]).

protects(Component,[Resource|OtherResources]) :-

(contain(Component,Resource),ensure_protect(Component,Resource) ;


```
not contain(Component,Resource)),
protects(Component,OtherResources).
match(Component) :- consider(Component),
require_protect(Resources),
protects(Component,Resources).
```

The above prolog predicate will do three basic functions,

- Protect the resources specified by the system Integrator to the developer in a component.
- Identify those resources in a component that need not to be protection.
- Match protected resources when searched.

This will now address the basic aim of this research and the purpose for which the AMA is developed. The results of the above predicate when implemented will be of the following format when NO resources are protected.

A.USER	COMMAND	USER	DB	AB
DB	READ	DB	Acs	Acs

In case of access control guards implemented the result obtained will be

A.USER	COMMAND	USER	DB	AB
DB	READ	DB	Acs	Deni

DB- Authorized User
AB – Unauthorized User
Acs – Accessed the record
Deni – Denied

In the second case where the resource is protected, the access is denied to the unauthorized user AB and when No guards are implemented then both the authorized user DB and the unauthorized user AB are able to access the information.

7. Conclusion

The basic aim of the proposed method is to develop an effective form of access control that stands out from the conventional forms of access control methods that are usually implemented in software organizations. Besides being efficient, easy to implement and very safe to use, the proposed model possesses code reusability that speaks for CBD properties. The results of AMAshown have proved that the access control guards are successful in blocking unauthorized access to information. The cost effectiveness of this model is established by the fact of code reusability in Uniframe platform. This method can also be further developed to account for component specifications that permit access control in searching. The researcher is involved in the process of developing similar type of matching algorithm that accounts for concrete component specifications. Further testing the system functionality with respect to access control properties can be created as a part of future work in continuation to the research done in this paper.

References

1. Abendroth, J., Jensen, C. *A Unified Security Framework for NetworkedApplications*, Proceedings of the 2003 ACM symposium on Applied Computing, pp351-357, Melbourne, Florida, 2003.
2. Alexander M. Crespi *An access control model for Uniframe Framework*, M.S. Thesis, Department of Computer & Information Science, Indiana University Purdue University Indianapolis, May 2005.
3. Brahnmath, G. *The UniFrame Quality Of Service Framework*, M. S. Thesis,Department of Computer & Information Science, Indiana University Purdue University Indianapolis, December 2002.
4. Burt, C., Bryant, B., Raje, R., Olson, A., and Auguston, M. , *Model Driven Security: Unification of Authorization Models for Fine-Grain Access Control* Proceedings of EDOC 2003, The 7th IEEE International Enterprise Distributed Object Computing Conference, Brisbane, Australia, September 16-19, 2003.
5. Huang, Z., *The UniFrame System Generative Programming Framework*, M.S. Thesis, Department of Computer & Information Science, Indiana University Purdue University Indianapolis, April 2003.
6. Harrington, A., Jensen C. *Cryptographic Access Control in a Distributed File System*, Proceedings of the eighth ACM symposium on Access control models and technologies, pp 158-165, Como, Italy, 2003.
7. Katzke, S. *Future Directions of the Common Criteria (CC) and the CommonEvaluation Methodology (CEM)*, National Institute of Standards and Technology, 2002.
8. Khan, K., Han, J. *Security Characterisation Framework for Trustworthy Component Based Software System*, Technical Report No. CIT/35/2003, University of Western Sydney, School of Computing and Information Technology, 2003.
9. Minsky, N., Ungureanu, V. *Unified Support for Heterogeneous Security Policies in Distributed Systems*, Proceedings of the 7th USENIX Security Symposium,Berkeley, California, 1998.
10. Lamport, L. "Specifying Systems: The TLA+ Language and Tools forHardware and Software Engineers

11. Object Management Group (OMG), *Model-Driven Architecture: A Technical Perspective*, Technical Report, OMG Document No. ab/2001-02-01/04, February 2001.
12. Object Management Group (OMG), *Resource Access Decision Facility Specification*, Technical Report, 2001.
13. Raje, R. *UMM: Unified Meta-Object Model for Open Distributed Systems*, Proceedings of 4th IEEE International Conference on Algorithms and Architecture for Parallel Processing, ICA3PP 2000, pp 454-465, Hong Kong, 2000.
14. Raje, R., Auguston, M., Bryant B., Olson, A., Burt, C. "*A Unified Approach for the Integration of Distributed Heterogeneous Software Components*, Proceedings of the 2001 Monterey Workshop Engineering Automation for Software Intensive System Integration, pp 109-119, Monterey, California, 2001.
15. Sandhu, R., Coyne E., Feinstein, H., Youman, C. "*Role-Based Access Control Models*, IEEE Computer, vol. 29, no. 2, pp 38-47, 1996.
16. Robert, S. *Concepts of Programming Languages*, Addison Wesley, 2002.

About The Authors

Ms. Dhowmya Bhatt received her B.Tech degree in IT in the year 2003 from M.K University and M.Tech in CS & IT in 2005 from M.S. University and was awarded as the "outstanding student" M.tech batch. she started her career with the IT industry and later switched to teaching and presntly has an experience of six years. She has authored many research papers in various International journals. Currently she is pursuing her doctoral degree and her research interests are Network security and access control.

Ekata Gupta is a doctoral degree holder and an Associate Professor in Krishna Institute of Engineering and Technology with a decade long career in the Teaching. She has participated in various National and International conferences all across India and has shared her research ideas. Dr. Ekata has authored above 30 research papers and her current area of interest is Neuro – Fuzzy, a comparison and combination of Neural Network and Fuzzy concepts. Her passion for research has taken her places and she has held various posts in her carrier. She is presently guiding about 5 research scholars who are pursuing their doctorate degree.

