2010

Pair-Wise Privilege Control for Cross-Domain Private Data Sharing

Jian Zhong
RMIT University, jian.zhong@rmit.edu.au

Peter Bertok
RMIT University, peter.bertok@rmit.edu.au

Zahir Tari
RMIT University, zahir.tari@rmit.edu.au

PAIR-WISE PRIVILEGE CONTROL FOR CROSS-DOMAIN PRIVATE DATA SHARING

Jian Zhong, Computer Science and Information Technology, RMIT University, Melbourne, Australia, jian.zhong@rmit.edu.au

Peter Bertok, Computer Science and Information Technology, RMIT University, Melbourne, Australia, peter.bertok@rmit.edu.au

Zahir Tari, Computer Science and Information Technology, RMIT University, Melbourne, Australia, zahir.tari@rmit.edu.au

Abstract

Enterprise-scale organizations have large numbers of internal and external users, with different privilege requirements spanning across many resources. The dynamic nature of modern organizations demands that they efficiently and securely provision and deactivate data privileges to reflect rapidly changing user responsibilities. Previous approaches to consolidated user provisioning have focused on constructing and maintaining a formal model of user privileges, in order to predict what role/roles should be assigned to any given user, based on user classification and other user attributes. In real-world deployments, formal models have not scaled well, because many users are unique and consequently there is no leverage to be gained by grouping them into roles. This paper proposes a scheme for dual control of user granular privilege and dynamic granular data access. The framework includes a correlated privilege control model and a label-based dynamic access level process. The method supports user activity control over cross-domain objects with variable data access granularity. It encompasses the advantages of existing role based and label based control, while reducing computation complexity and storage requirements. The proposed method has been formally verified and implemented in JAVA.

Keywords: Privilege control, Role-based, Granular data control, Sensitive and private data sharing, Multi-domain.

1 INTRODUCTION

Diverse organizations, governments and individuals share vast amounts of information. Since information sharing can potentially harm certain parties, it is typically governed by roles and policies that support *subject privilege control* and *data granularity control* (Lei Zhang et al. 2008). Both of these methods have been well studied and, typically, they are enforced by role-based access mechanisms and label based approaches, respectively.

Within an organization, roles are generated for various job functions. Specific roles are assigned in accordance with the permissions to perform certain operations. The users acquire permissions based on their roles rather than generality, and as such the rights of individual user management becomes the assignment of appropriate roles to the user (Ferraiolo, D.F. and Kuhn, D.R., 1992, Sandhu, R. et al. 1996). Also, granular data control, or fineness with which data fields are sub-divided addresses two main aspects- (i) How to achieve fine-grained granularity and data selection/sanitization ? and (ii) object-based privilege control. The latter will be discussed in this paper.

With an increasing complexity of infrastructure along with a growing number and diversity of users who must access it, the apparent limitation of privilege control on which traditional role is based has emerged. Furthermore, providing data granularity adds difficulty to the deployment of existing approaches in circumstances where the majority of users are unique i.e. their role is assigned to a single user only.

The main issue is to control two aspects simultaneously: (i) When accessing the whole object, which activities are available for each granular data? and (ii) For numerous unique users, how to realize dynamic granular data access level control. We will be examining three sub issues here- (i) *unique subjects*, (ii) *collaboration and (iii) multi-domain application*.

In this paper, we propose a solution to integrate sophisticated access control into a request-based pair-wised privilege control model, which has three modules – (i) parameterized 3D subject granular privilege control (ii) object-based dynamic granular data control and (iii) privilege refinement. The model will consider the advantages of the existing role-based and label-based control. The rest of this paper is organized as follows. In section 2, we provide a general background dealing with the fundamentals of the cooperation between user privilege granularity controls and object dynamic granular data level control. It also reviews the previous work by comparing it to the contemporary solutions. This is followed by a detailed proposed method, containing 3D user privilege granularity control, object dynamic granular data level control and their collaboration. Conclusions are provided at the end of the paper.

2 LITERATURE REVIEW

There have been many attempts considering unique users with diversity privileges (Qingfeng He and Annie I Anton 2003; Xueli Li et al. 2005; Ali E. Abdallah and Etienne J. Khayat 2005). By incorporating context tables, *one solution* can support finer-grained privileges and subject variable request (Qingfeng He and Annie I Anton 2003). With additional condition list, *another existing approach* offers complex roles with diverse privileges (Xueli Li et al. 2005), while in the *third approach*, the traditional concept of 'role' fades out is replaced by a new parameterized model, which not only supports unique users better but also reduces the storage consumption than traditional role-based approaches (Ali E. Abdallah and Etienne J. Khayat 2005). Nevertheless, these solutions try to add different modules to extend additional privileges with which the users can achieve diversity accordingly. Whereas, decreasing efficiency is unpreventable as the unique users come and go continuously. In addition, in (Ali E. Abdallah and Etienne J. Khayat 2005), role models are difficult to be built before the users lodge their requests. In contrast to these solutions, the proposed method builds an enhanced privilege assignment model that can directly assign privilege to subjects by request, priority check and privilege refinement. The storage consumption decreases when storing

diverse roles on the subject server becomes unnecessary. Moreover, the *delicate modules* and components offer efficient privilege assignment.

For collaboration control, connecting subject with object-based privilege control is considered in previous approaches (Acevedo M.T. et al. 1997; Qingfeng He and Annie I Anton 2003). However, the former solution overlooks the granular data control like the latter and it provides a non-independent object based privilege control as well as limits the dual control performance. In the proposed solution, same hierarchical models, for subject and object based privilege control, add efficiency to granular data control. In addition, an independent object controller offers high performance and supports special condition control specifically, say, the worldwide-scale enterprise. For multi-domain application, building a roaming table (Lorenzo D. Martino et al. 2008) that explores the method to a great extent without referring to the practical issues of role changing and data granularity will not only consume considerable storage resource but also lower the privilege assignment efficiency. In contrast, a dynamic hierarchy component that caters for data roaming without compromising a high management cost is adopted by the proposed method.

3 PROPOSED METHOD

In this section, the proposed pair-wise privilege control scheme is examined from three aspects: namely the Privacy label based Subject granular Privilege Control (PSPC) module; Privacy label based Object granular Data Control (PODC) module and the collaboration control model (See Figure 1).

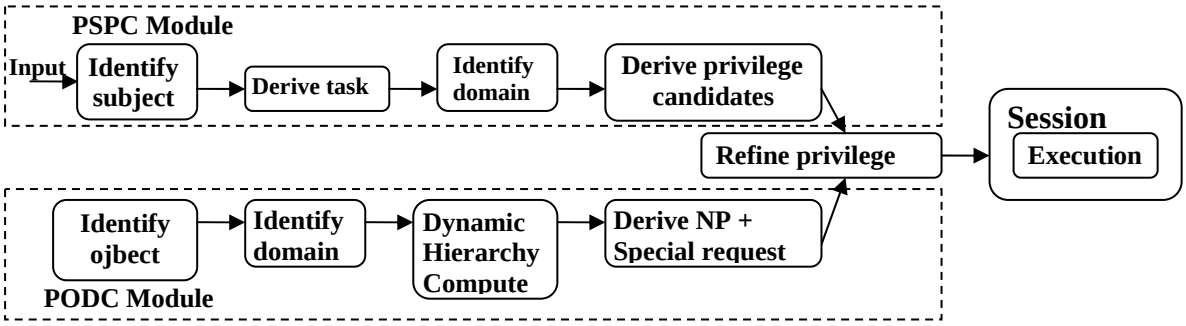

Figure 1. Pair-wise Privilege Control Model

Basic concepts and definitions are given below:

Definition 1: A set of subject $S = \{S_i \mid i = 1, 2, \cdots, n\}$, where *n* is the number of subjects in a data sharing environment. A set of object $O = \{O_i \mid i = 1, 2, \cdots, n\}$, where *n* is the number of objects. Each object has a set of granular data $O_x = \{OGD_i \mid i = 1, 2, \cdots, n\}$, where *n* is the number of granular data. **Definition 2**: A set of subject activity $SA = \{SA_i \mid i = 1, 2, \cdots, n\}$, where *n* is the number of subject activity. A set of *Subject Activity level* represented by $SL = \{\Omega(SA)_i \mid i = 1, 2, \cdots, n\}$ denotes the subject activity priority on the required object, where *n* is the number of subject activity level and operation Ω(*SA*) denotes the sub set of *SA*. *Subject Grade* denotes the subject's overall privacy priority, which is represented by *SG*. A set of Subject Sub Grade $SSG_x = \{\Omega(SL)_i \mid i = 1, 2, \cdots, n\}$ denotes the access priority for each granular data of an object, where x denotes a certain sub grade and *n* is the number of activity levels included in the sub grade.

Definition 3: *Object Grade* (OG) denotes the minimum overall privacy priority of the requiring subject. A set of object sub grade $OG = \{OSG_i \mid i = 1, 2, \cdots, n\}$, where *n* is the number of granular data and OSG denotes Object Sub Grade. A set of Negative Permission $NP = \{NP_i \mid i = 1, 2, \cdots, n\}$, where *n* is the number of negative permission. **Definition 4**: A set of Duty $D_x = \{(S_i, \Omega(SA)_j, O_k)\}$,

which indicates duty *x* composing of subject *i* requires access on object *k* with a set of activity *j*. Permission $P_x = \{(D_i, Boolean)\}$, where *Boolean* indicates acceptance and rejection.

3.1 PSPC Module

The PSPC is a subject based privilege control module and has three components which are core PSPC, hierarchical PSPC and Subject Privacy Label Generator.

3.1.1 Core PSPC

The core defines five basic elements in PSPC module, which are subject (S), duty (D), operation/activity (A), object/resource (O), privacy label (SPL) and session SYN (SYN(s)). *Operation* is a set of subject activities. Subject operation is presented as subject activity. *Privacy label* is a control frame containing processing parameters and subject privilege candidates. Session SYN is a parameter indicating a valid time period for the refined privileges. In PSPC, no negative permissions are assigned to SPL, while it is to be assigned only to Object Privacy Label. PSPC is a request-based subject privilege strategy. After a subject duty request is sent to subject server, it will be first checked whether the subject is allowed to lodge the duty request. The duty request is derived from Hierarchical PSPC component and the privilege candidates that are included in SPL.

3.1.2 Hierarchical PSPC

Hierarchical PSPC defines the control of subject granular privileges. **Rule 1**: *Subject Activity* in the same *Subject Activity Level* is mutually exclusive, which means only one will be activated in the same session, while these activities can work on granular data in the same session if they are on different levels. **Rule 2**: *Subject Activity Level* in one *Subject Sub Grade* must be different to other activity levels in the same subject grade. **Rule 3**: Different *Subject Sub Grades* are controlled independently. **Definition 5**: Subject Activity Level correlation process operation is Θ which is an arithmetic operation set of subject levels. If the process result of SL_1 and SL_2 is needed for the further use, the operation result will be $\Theta R_{S_A(L_1, L_2)}$.

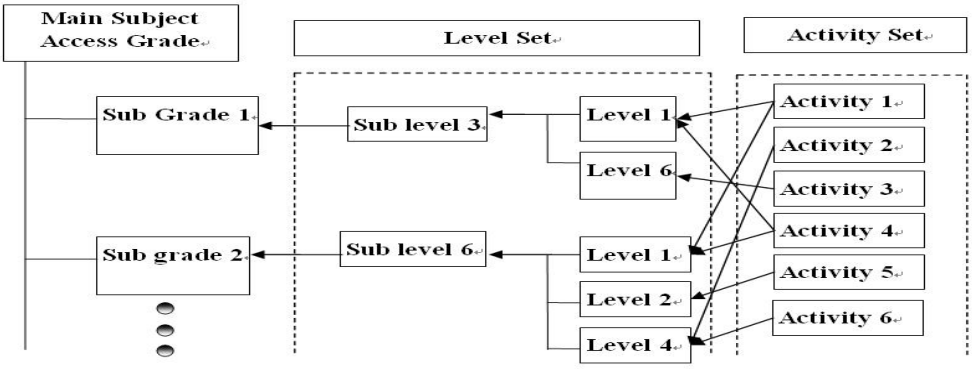

Figure 2. PSPC Hierarchy Structure

Figure 2 shows an example of PSPC hierarchy structure, from which we can derive $SG = \{SG\_val, (SSG_1, SSG_2)\}$, where *SG_val* denotes the value of *SG*; $\Omega(SL)_3 = \{SL_1, SL_6\}$ and $\Omega(SL)_6 = \{SL_1, SL_2, SL_4\}$; $SL_1 = \{SA_1, SA_4\}$ and $SL_2 = \{SA_5\}$ etc. Basically, after a request has been lodged to a subject server, the subject is to be associated with an overall privacy priority, also called main subject access grade and represented by *Subject Grade* (SG). The *SG* usually consists of a set of *SSG* which are associated with granular data of the required object. Each *SSG* is composed of a set of *SL*s, which indicates the priority and correlation between different *SA*. *SA*s included in the *SL* are the behaviours indicating what can be executed over the object.

3.1.3 Subject Privacy Label Generator

After the processing of component of Hierarchical PSPC, the construction of the parameterized SPL is generated as. SPL (subject) = [subject identity (hash), domain classification code, object identity (hash), flags, PSPC control code, session SYN] Where *domain classification code* (DCC) indicates the activated working area; *flags* indicates special structure of the frame. Normally, it should be zero and it is designed for future works; PSPC control code (SCC) contains the semantic PSPC Hierarchy Structure, which can be represented as follows:

$SCC = \{SG\_val\} \cup \{SSG_x\_val \mid x \in [1, n]\} \cup \{\Omega(SL)_y\} \cup \{\Omega(SA)_z\}$; where n denotes the number of required granular data on an object; the number of y and z depends on subject request.

3.2 PODC Module

The Pair-wise Object granular Data Control (PODC) module is a label based control frame employed for dynamic granular data and multi-domain application. The core PODC defines three basic elements in PODC module, which are meta-data, dynamic hierarchy and object privacy label. *In order to explain the paper further, we define the following terms: Granular data* – is fine-grained meta bits. *Dynamic hierarchy* - changeable object access priority control, which is detailed in Dynamic Hierarchy. *Privacy label* - control codes derived from *Hierarchy Assignment* (HA) which denotes the computation of connecting relative access priority to applied privacy priority. *Data mapping* - operation of making connection between granular data and their access priority.

3.2.1 Dynamic Hierarchy

Component *Dynamic Hierarchy* has two main functions, which are Hierarchy Assignment (HA) and Condition Assignment (CA). When data is roaming to other domains, the original privacy priority control may not be able to adjust properly. Figure 3 shows the concept of HA applied mapping.

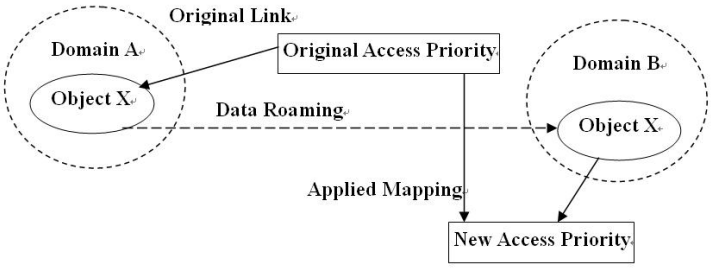

Figure 3. Hierarchy Assignment (HA)

In Figure 3, Access Priority is represented by *OG* including *OSG*s. When the object X (including granular data) are created, it is processed by the original object server in the domain A, where original access priority is a non-absolute classification value. If such a data is required for roaming, the original object server will first assign a *basic sub grade* (BSG) to one or more granular data bit(s). Then, all other data bits can be assigned proper grades referring to BSG at original object server. When the data is roaming to domain B, the object server in domain B will first map BSG to a basic applied grade (BAG) based on the data sharing environment and sharing requirement. Then, the system maps the rest original granular data sub grades to applied grades based on their correlation to the BSG. PODC aims to achieve object dynamic granular data access control.

3.2.2 Object Privacy Label Generator

After the object server receives the request, the data is prepared for sharing, with object privacy label (OPL) attached. The OPL is generated by collecting the control info from Dynamic Hierarchy component.

OPL (object) = [object identity (hash), domain classification code, subject identity (hash), flags, PODC control code, session SYN]. The most fields are similar as SPL while PODC control code (OCC) contains the semantic PODC Dynamic Hierarchy Structure, which can be represented as follow.

$$OCC = \{OG\_val\} \cup \{OSG_x\_val \mid x \in [1, n]\} \cup \{\Omega(NP)_y\} \cup \{\Omega(SP)_z\}$$

Where *n* denotes the number of required granular data on an object; the number of *y* and *z* depends on the object owner/manager's requirement.

3.3 Pair-wise Privilege Control Model

The Pair-wise Privilege Control Model introduces a collaborating structure to assist in defining and carrying rules of multi-domain data sharing for large amount of unique subjects' environments. It enforces the concept of dual control both subject granular privilege and object granular data. For sharing information in such an environment, each subject sends request with duties including targets and activities to subject server, then derives SPL with privilege candidates. After that, object server will generate an OPL with authorized granular data. Finally, the component of privilege refinement computes out allowed privileges and sends them to a valid session (See Figure 1).

3.3.1 The Component of Privilege Refinement

Basically, after both privacy labels have been produced, approval privileges will be calculated by the component of privilege refinement, which has two parts: a) validation check of both subject and object general info, and b) privilege refinement. The former is detailed as follows and the later is in the following section.

$$SPL \oplus OPL = (SPL.subject\_id \oplus OPL.subject\_id) \,\&\, (SPL.DCC \oplus OPL.DCC) \,\&$$
$$(SPL.object\_id \oplus OPL.object\_id) \,\&\, (SPL.sessionSYN \oplus OPL.sessionSYN) +$$
$$RE(SPL.PSPC, OPL.PODC)$$

Where symbol $\oplus$ indicates the data matching process and & denotes logic 'and'; *RE* denotes Refinement Algorithm. If one of the terms is false, the overall result is false and the *RE* function aborts. Otherwise, the results of *RE* function will output to a valid session.

3.3.2 Refinement Algorithm (RE)

The *RE* function computes proper permissions for the subject request. Figure 4 (LHS) describes the first step of permission computing, which outputs the overall permission of the access. If the subject's grade is equal to or higher than object prohibited grade, authorization will be given and move to the next step, otherwise access is denied. $S_iPL.S_iG$ denotes the Subject Grade of Subject i; OPL.OG denotes object overall grade. The algorithm for sub grade computing is given in Figure 4 (RHS). The process compares all the subject sub grades to all object granular data grades respectively, where → denotes move to the next. Also, we have notations of permission symbols and extended *SL* operations detailed as follows:

```
function Compare (subject i 's grade, object grade)
    (sg = SiPL.SiG)→(og = OPL.OG)
    if sg ≥ og do
        return Si =ΓO
        move Next Process
    else
        return Si !Γ O
        end
    end if
```

```
function Compare (all subject sub grades, all object granular data grades)
    for all x such that 1 ≤ x ≤ m do
        if SiSGx ≥ OSGx then
            return SiSGx =Γ OSGx
        else
            return SiSGx !Γ OSGx
        end if
    end for
    move Next Process
```

Figure 4. Grade Computing and Sub Grade Computing

Notion 1: (Positive Permission Operation =Γ) $S_i SG_x = \Gamma^{SA_j} OSG_y$ denotes subject *i* is allowed to have activity *j* on the object granular data *y* in its duty *x*. **Notion 2**: (Negative Permission Operation !Γ) $S_i SG_x !\Gamma^{SA_j} OSG_y$ denotes the object's granular data y has negative permission (not allowed to access) on subject *i* with activity *j* in duty x. **Notion 3**: (Activity and Level Process Θ)

```
function Compare (SiL)
    for all x such that 1 ≤ x ≤ m do
        for all y such that 1 ≤ y ≤ n do
            if Θ→ then
                return SiLy.precondition(SiLx)
                continue
            else if Θ← then
                return SiLx.precondition(SiLy)
                continue
            else if Θ↔ then
                select SiLx or SiLy
                return selection
                continue
            else if Θ↕ then
                continue
            else continue
            end if
        end for
    end for
```

```
function execution (SiB)
    Compare (subject i 's grade, object grade)
    Compare (all subject sub grades, all object granular data grades)
    Compare (SiL)
    for all x such that 1 ≤ x ≤ m parallel do   //for all sub grades
        select executing activity
        if selection is in negative permission list then
            activity denied
            continue
        end if
        if precondition = true then
            select precondition activity or abort
            execute selection
            continue
        end if
        if mutually exclusive = true then
            select one of the mutually exclusive activities or abort
            execute selection
            continue
        end if
        execute selection
    end for
```

Figure 5. Level Computing and Activity Execution

$S_i L_x \Theta^{\rightarrow} S_i L_y$ denotes for the subject *i* and a granular data of the object, level y must be processed after level *x*; $S_i L_x \Theta^{\leftarrow} S_i L_y$ denotes for the subject i and a granular data of the object, level y must be processed before level *x*.; $S_i L_x \Theta^{\leftrightarrow} S_i L_y$ denotes for the subject i and a granular data of the object, level y and level *x* are mutually exclusive, which means only one level will be processed.; $S_i L_x \Theta^{\updownarrow} S_i L_y$ denotes for the subject i and a granular data of the object, level y and level *x* can be processed simultaneously. The process in Figure 5 (RHS) is activity execution, which describes each required activity included in each level that is executed based on the results of the three algorithms explained above.

Based on the algorithms, collaborating control of subject granular privilege and object dynamic granular data has been achieved. The detailed processing algorithms are provided in an extended thesis. We have verified our proposed scheme by Failures-Divergence Refinement (FDR) (Goldsmith, M. 2005), a model verification tool based on Communicating Sequential Processes (CSP) state

machines. The proposed scheme is implemented in JAVA and the detailed verification and implementation are described in a separate thesis.

4 CONCLUSION

This paper has proposed a mechanism for collaboration of subject granular privilege control and object dynamic granular access level control. It addresses both the appearance of subject privilege granularity control in plentiful unique subjects and its connection with a changeable granular data permission control. The model can be implemented in diverse circumstances in which the private and sensitive data sharing is practical. Numerous unique subjects in such system will not consume as much memory overhead as traditional role-based methods. Dynamic granular data privilege control is supported by the proposed mechanism.

References

Acevedo, M.T. Fillingham, D. Nicolettos and J.L. (1997). Enterprise security application of partition rule based access control (PRBAC). In IEEE 0-8186-7967-0/97.

Ali E. Abdallah and Etienne J. Khayat. (2005). A formal model for parameterized role-based access control. In IFIP International Federation for Information Processing, 173/2005, 233-246.

Elisa Bertino, Barbara Carminati and Elena Ferrari. (2001). XML security. In Information Security Technical Report, Vol 6, No.2, 44-58

Elisa Bertino, Silvana Castano and Elena Ferrari. (2001). Securing XML documents with Author-X. In IEEE Internet Computing, Data Security, 1089-7801/01

Ferraiolo, D.F. and Kuhn, D.R. (1992). Role-based access control. The 15th National Computer Security Conference, 554–563.

Goldsmith, M. (2005). FDR2 User's Manual Version 2.82. Formal System (Europe) Ltd.

Hitachi ID Systems, Inc. (2009). Beyond Roles: A Practical Approach to Enterprise User Provisioning. Available at http://www.idsynch.com/docs/beyond-roles.html

Kim, I. G., and Choi, J. Y. (2004). Formal verification of PAP and EAP-MD5 protocols in wireless networks: FDR model checking. In Advanced Information Networking and Applications, AINA, 18th International Conference on, 2.

L. Giuri and P. Iglio. (1997). Role Templates for Content-based Access Control. In Proceedings of the 2nd ACM Workshop on Role-based Access Control. Fairfax, Virginia, USA, 153-159

Lei Zhang et al. (2008). A Framework for Maximizing Utility of Sanitized Documents Based on Meta-labelling. In 2008 IEEE Workshop on Policies for Distributed Systems and Networks, pp. 181-188

Lorenzo D. Martino, Qun Ni, Dan Lin and Elisa Bertino. (2008). Multi-domain and privacy-aware role based access control in eHealth. Computer Science; Purdue University, USA

Qingfeng He and Annie I Anton. (2003). A framework for modelling privacy requirements in role engineering. In Department of Computer Science, North Carolina State University, Raleigh, NC 27695-8207, USA

Sandhu, R., Coyne, E.J., Feinstein, H.L. and Youman, C.E. (1996). Role-based access control models. IEEE Computer (IEEE Press) 29 (2), 38–47.

T. Jaeger, T. Michailidis and R. Rada. (1999). Access Control in a Virtual University. In Proceedings of the 8th International IEEE Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, California, USA, 135-140

Xueli Li, Nomair A. Naeem and Bettina Kemme. (2005) Fine-granularity access control in 3-tier laboratory information systems. In Proceedings of the 9th International Database Engineering and Application Symposium (IDEAS '05).