"Another reason for the existence of multiple data sources is that not all relevant data is controlled or maintained by the enterprise. Examples of such external information sources include schedules, weather reports, stock prices, phone directories, parts catalogues, and a rapidly growing number of public databases. Any of this information could be important to an enterprise and all of it is available on-line."

Tim Connors and Peter Lyngbaek
Providing Uniform Access to Heterogeneous Information Bases [CON88].

"This is based on the fact that if you trust someone, you believe that he won't embarrass you. Note that if someone deceives you, nothing can be done as long as that fact has not come to light. However, as soon as you detect that fact, you can revoke his rights."

M.L. Kersten and W. Seminarium
Privacy and Security in Distributed Database Systems [KER82].
6.1 Introduction

In chapter 4, we have discussed our proposed access control mechanism in DHDBMS. We have mentioned in Chapter 1, that work is in progress to develop several DHDBMS prototypes in different organizations across the globe. Some of these prototypes were discussed in chapter 2 along with their architectures. In this chapter, we will discuss the applicability of our proposed access control mechanism to these DHDBMS prototypes.

In chapter 3, we have discussed an access control mechanism for a prototype distributed homogeneous database management system R*. We also discussed that this mechanism cannot be used for DHDBMS because a DHDBMS operates on a Global Data Model concept and its component DBMSs may have been built using different data models. In chapter 3, we also discussed an access control mechanism for a DHDBMS proposed in [WAN87]. In the present chapter, we will compare our proposed solution to the solution proposed by [WAN87].

6.2 Applicability of Access Control Mechanism to Existing DHDBMS Prototypes

We have discussed in chapter 4, that the proposed access control mechanism is built in the global model. The proposed access control mechanism is based on view mechanism and application program authorization technique. We have also discussed that a protection view in a DHDBMS is created in the Global Functional System and it is defined over the
data objects of Global Conceptual Schema. Similarly, an application program is also installed in Global Functional System and it is also based on the data objects defined in the Global Conceptual Schema. We have also discussed query processing technique and execution of an application program in a DHDBMS.

It is clear from the discussions in chapter 4, that by authorizing views and application programs, we can achieve content-dependent and functional access control in a DHDBMS. It is also clear that this access control mechanism is built only in the Global Functional System, and there is no change to be incorporated in the preexisting operational component DBMSs. We have not proposed any change in the access control mechanism (if any) of the component DBMSs. We have also not proposed any change in the local query processing techniques of the component DBMSs. Our proposed access control mechanism is simple and can be easily implemented at the global level. We have proposed the traditional view materialization technique at the global level. We have simply suggested a small change during distribution of subqueries to different sites. We have proposed that the User Identification Number and other identification information of a user should be replaced by the User Identification Number and corresponding identification information of the global DBA in a subquery. This will help in accessing data objects from component DBMSs.
In chapter 2, we have discussed some DHDBMS prototypes. All these prototypes have powerful data manipulation languages. They have facilities of defining views and running of application programs. An authorization mechanism will have to be built in their global models. We do not foresee any problem in introducing access control mechanism in their Global Functional System. It is felt that our proposed access control mechanism can be built in all the DHDBMS prototypes discussed in chapter 2.

It should be noted that as per our proposed mechanism all the accesses to component DBMSs will be done by the global DBA (on behalf of various users) and therefore accounting of local system usage will be done against the global DBA by all the component DBMSs. Therefore, it is necessary to build an accounting system at the global model, which keeps track of local system usage done by global DBA on behalf of different users. It may also be desirable to keep audit trail of such transactions so that it could be found out later on (if required) that which user has requested for which information/data from the component DBMSs. These facilities can be implemented in the system without much difficulty.

6.3 Comparative Assessment

In this section we compare our proposed access control mechanism in a DHDBMS with the proposed solution of [WAN87]. This is presented in a tabular form in Table 6.1.
1. The access control mechanism is proposed for a DHDBMS architecture based on the ER model. In this architecture, the component databases supported are built using the Relational, Hierarchical and Network data models. It may be difficult to use this solution for a DHDBMS architecture based on other data models.

2. The [WAN87] solution is for content-dependent access control. The proposed solution is for content-dependent and functional access control.

Table 6.1: Comparative Assessment with [WAN87] Solution
Table 6.1 (Contd..)

<table>
<thead>
<tr>
<th>[WAN87] solution</th>
<th>Proposed solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>3. The [WAN87] solution follows a modified view materialization algorithm.</td>
<td>The proposed solution follows the traditional view materialization algorithm.</td>
</tr>
<tr>
<td>4. The [WAN87] solution needs a particular query processing technique. This technique is complex.</td>
<td>The proposed solution supports the traditional query processing technique. It replaces user ID etc. of a user by the ID etc. of the DBA in the subqueries during distribution of subqueries to various sites. This technique is simple.</td>
</tr>
<tr>
<td>5. When a view is defined at the global level, some operations are performed in the relevant component databases.</td>
<td>When a view is defined at the global level, no operations are performed in the component databases.</td>
</tr>
</tbody>
</table>
6. In some applications, there may be short hand views [TEM87b]. The [WAN87] solution will materialize these views also in the same way as protection views and therefore will consume lot of system resources. If these short hand views are referred frequently, it will have lot of overheads because of

(i) Creation of views
(ii) materialization of views, and
(iii) deletion of views at the level of component DBMSs.

7. Because of typical view materialization algorithm and query processing technique, overheads are high.

The proposed solution replaces view names in the query by global data object names in the Global Functional System and then processes the intermediate query in the normal way. Therefore, these overheads will not occur.

---

Table 6.1 (Contd..)

<table>
<thead>
<tr>
<th>[WAN87] solution</th>
<th>Proposed solution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The proposed solution replaces view names in the query by global data object names in the Global Functional System and then processes the intermediate query in the normal way. Therefore, these overheads will not occur.</td>
</tr>
</tbody>
</table>
It is clear from the table that our proposed solution is a
general solution and it is more efficient and simpler than
the solution proposed in [WAN87]. Also our solution
provides Content-dependent and Functional access control
while the solution presented in [WAN87] is for Content-
dependent access control only.

6.4 Summary

In this chapter we have discussed the
applicability of our proposed solution to different DHDBMS
prototypes. We have shown that our proposed solution is
simple and can be implemented without any difficulty in
various DHDBMS prototypes. We have also compared our
proposed solution with the solution proposed by [WAN87]. We
have shown that our proposed solution is simple, efficient
and general.