[LDS90] Lunt, T. F., D. E. Denning, R. R. Schell, M. Heckman, and W.R. Shockley. *The Seaview security model.* IEEE Transactions on Software Engineering, Vol 16, pp 593-607, 1990.

[NG92] Nair, Sunil and Shashi K. Gadia. *Algebraic optimization in a relational model for temporal databases,* Proc. First International Conference on Information and Knowledge Management, pp 169-176, 1992.

[PMP94] Pissinou, Niki, Kia Makki, and E. K. Park. *Towards a framework for integrating secure models and temporal databases*. Proc. of Third International Conference on Information and Knowledge Management, 1994, pp 280-287.

[SW92] Smith K. and M. Winslett. *Entity modeling in the MLS relational model.* Proceedings of Eighteenth VLDB, pp 199-210, 1992.

[Ta+93] Tansel, Abdullah, et al, Eds. *Temporal Databases: Theory, Design, and Implementation*. Benjamin/Cummings, Redwood City, California, 1993, pp 28-66.

[WSQ94] Winslett, Marianne, Kenneth Smith, and Xialei Qian. *Formal query languages for secure relational databases.* ACM Transactions on Database Systems, Vol 19, 1994, pp 626-662.

[DLS87] D. Denning, T. Lunt, R. Schell, et al. *A multilevel relational data model.* Proceedings of IEEE Symposium on Security and Privacy, pp 220-234, 1987.

[DLS88] D. Denning, T. Lunt, R. Schell, W. R. Shockley and M. Heckman. *The Seaview security model.* Proceedings of IEEE Symposium on Security and Privacy, pp 1988.

[Ga86] Gadia, Shashi K. *Weak temporal relations*. ACM Transactions on Database Systems, 13(4), pp 418-448, December 1988.

[Ga88] Gadia, Shashi K. *A homogenous relational model and query languages for temporal databases*. ACM Transactions on Database Systems, 13(4):418-448, December 1988.

[Ga97] Gadia, Shashi K. A *bibliography and index of our works on belief data: concept of error and multilevel security*. Tech. Report TR97-13, Computer Science Department, Iowa State University, Ames, 1997.

[GB89] Gadia, Shashi K. and Gautam Bhargava. *A formal treatment of errors and updates in a relational database*. 1988-89. An unpublished manuscript, available as Tech. Report TR97-14, Computer Science Department, Iowa State University, Ames, 1997.

[GC95] Tsz Shing Cheng and Shashi K. Gadia. *An algebra for belief persistence in multi-level security*. Version 1, 1995. An unpublished manuscript, available as Tech. Report TR97-16, Computer Science Department, Iowa State University, Ames, 1997.

[GC96] Tsz Shing Cheng and Shashi K. Gadia. *An algebra for belief persistence in multi-level security*. A revised version of [3] incorporating some findings from [4]. Version 2, 1996. An unpublished manuscript, available as Tech. Report TR97-18, Computer Science Department, Iowa State University, Ames, 1997.

[GN93] Gadia, Shashi K. and Sunil Nair. *Temporal databases: A prelude to parametric data*. In [Ta+93], pp 28-66.

[GQ95] Gong, L. and X. Qian. *Enriching the expressive power of security labels.* IEEE Transactions on Knowledge and Data Engineering, pp 839-841, Vol 7, 1995.

[GY88] Gadia, Shashi K. and Chuen-Sing Yeung. *A generalized model for relational temporal databases.* ACM SIGMOD International Conference on Management of Data, 1988, pp 251-259.

[GY91] Gadia, Shashi K. and Chuen-Sing Yeung. *Inadequacy of Interval Timestamps in Temporal Databases.* Information Sciences, Vol 54, pp 1-22, 1991.

[HOT91] Haigh. J., R. O'Brien, and D. Thomsen. *The LDV secure relational DBMS model.* Database Security IV, pp 265-279, 1991.

[JS90] Jajodia, S. and R. Sandhu. *Polyinstantiation integrity in multilevel relations.* Proceedings of IEEE Symposium on Research in Security and Privacy, pp 104-115, 1990.

[JS91] Jajodia, S. and R. Sandhu. *Toward a multilevel secure relational data model.* Proceedings of ACM-SIGMOD, pp 50-59, 1991.

simpler query language than that in the WSQ model. In [GY88,GN93] it has been shown that whereas the query languages in the parametric model handles the natural language constructs “or”, “and”, and “not” symmetrically, the languages that use tuple timestamps do not achieve this symmetry. The same arguments in [GY88,GN93] would reveal a lack of symmetry in the WSQ model as well as in other models in multilevel security literature.

Another advantage of the parametric model is that it leads to a seamless integration of ordinary, temporal, spatial, and belief data. This integration in the parametric model would be much tighter than the integration of temporal and multilevel security data in [PM94].

The identities in the parametric model and algebraic optimization has been discussed in [NG92]. That approach to algebraic optimization also applies to the parametric model for multilevel security.

Lastly it must be remembered that the data in the real world has more complex structure than 1nf. If our language seems simpler, it is because it has imitated the “real” structure rather then temper with it to fit the 1nf mold. In fact, the structure of multilevel security data is far more complex than what is covered in this paper in terms of u-polyinstantiation. Key-polyinstantiation represents the true form of polyinstantiation, and this form of polyinstantiation has been covered extensively in our works during the last decade. In particular, key-polyinstantiation in multilevel security has been covered in [CG97a,CG97b].

ACKNOWLEGMENTS. The author wishes to thank Suraj Kothari, Giora Slutzki, Akhilesh Tyagi, Marianne Winslett, and anonymous referees for their helpful comments in improving this paper.

REFERENCES

[BG89] Gadia, Shashi K. and Gautam Bhargava. *A 2-dimensional temporal relational database model for querying errors and updates, and for achieving zero information-loss.* Technical Report 89-24. Department of Computer Science, Iowa State University, Ames, Iowa, December 1989.

[BG90] Gautam Bhargava and Shashi K. Gadia. *The concept of an error in a database: an application of temporal databases*. Appeared in Proceedings of INSDOC COMAD’90 International Conference on Management of Data, December 1990. Also available as Tech. Report TR97-15, Computer Science Department, Iowa State University, Ames, 1997.

[BG93] Bhargava, Gautam and Shashi K. Gadia. *Relational database systems with zero information loss*. IEEE Transactions on Knowledge and Data Engineering, Vol 5, pp76-87, 1993.

[BL75] Bell, D.E. and L. J. LaPadula. *Secure computer systems: unified exposition and multics interpretation.* Tech Report MTR-2997, MITRE, 1975

[CS95] Chen, Fang and Ravi S. Sandhu. *The semantics and expressive power of MLR data model*. Proceedings of IEEE Symposium on Security and Privacy, 1988.

Example 15. Now we consider the query *list all names I do not believe exist but some lower users do*. In the two models this query is expressed as follows.

WSQ model:
```
(select Name from emp believed by anyone)
minus
(select Name from emp believed by self)
```

Parametric model:
```
select Name
from emp e
where me ∉ [[e]]
```

In the WSQ model the following SQL-like expression is mentioned, and it is stated that the expression will not work for the given query.

```
select Name
from emp
believed by anyone
where Name not in   (  select Name
                       from emp
                       believed by self)
```

As seen above, the SQL query for the parametric model is simpler and the user may not feel a need for such a complex expression form because in the parametric model the information about a single object resides in a single tuple. Even if such information resided in different tuples, query in SQL for the parametric model would be as follows, and the problem stated in [WSQ94] would not arise.

```
select e.Name
from emp e
where e.Name ↓ e.owner not in (  select e′.Name
                                 restricted to e.owner
                                 from emp e′
                                 where me ∈ [[e′]])
```

8. Conclusions

This paper has shown a fundamental relationship that exists between the parametric model and multilevel security databases. It has also shown how the parametric model for temporal data readily adapts to multilevel security. In this venture, the only changes in the temporal model are as follows:

- Change of the term *instant* (of time) to the term *user* (or *user level*)
- Change of the term *temporal element* to *user element*
- Derivation of user hierarchy in multilevel security as a special case of the user hierarchy in a generic parametric model.

An exhaustive comparison between the WSQ and the parametric models under u-polyinstantiation was given. It has been found that the parametric model leads to a

1. Note “emp e” creates alias e of the emp relation; this is not a cross product of emp and e

Example 10. The query *list all users who believe in the existence of John* is expressed in the two algebras as follows:

WSQ model: $\Pi_{Label}((\sigma(emp, Name = John) \times self) \uparrow anyone)$

Parametric model: $[\![\sigma(emp, Name = John)]\!]$

Example 11. The query *list the beliefs about John's department* is expressed in an SQL-like languages in WSQ and the parametric models as follows.

WSQ model:
```
select Dept, Label
    from emp, self
    believed by anyone
    where Name = John
```

Parametric model:
```
select Dept
    from emp
    where Name = John
```

Example 12. *List the names of all employees anyone believes to exist*.

WSQ model: Π_{Name} (emp) $\uparrow$ anyone

Parametric model: Π_{Name} (emp)

Example 13. Consider the query *list all names everyone believes to exist.* This query is expressed in the algebra of the WSQ model as follows.

WSQ model: Π_{Name} (emp) –
Π_{Name} (Π_{Name} (emp) $\times$ anyone – ((Π_{Name} (emp) $\times$ self) $\uparrow$ anyone)

The above expression is complex because it has to handle the quantifier "for all" ($\forall$) at the relation level. However, note that the English query does not involve quantification at the relational level, but only at the object level. Though relational level quantifications would be complex in the algebra for parametric model, the object level quantification would not. In the algebra of the parametric model it is expressed as follows.

parametric model: $\Pi_{Name}\, \sigma(emp, [\![\ ]\!] = \textit{Users},)$

Example 14. The query *list employee names believed to exist at my level but at no level below me* is expressed in the two models as follows.

WSQ model:
```
select Name
    from emp
    where Name not in ( select Name
                        from emp
                        believed by (select Label
                                     from anyone
                                     where Label not in ( select
                                                          from self)))
```

Parametric model:
```
select Name
from emp e[1]
where [[e]] = me
```

metric model, when a user poses a query, the query is executed for the whole database, and if the user wants to restrict the computation to a level u, every operand relation should be restricted to u by the user. On the other hand, in the WSQ model when a user u poses a query, it is evaluated for the data at level u. If the user wants to involve the data at additional levels, it should use "$\uparrow$ U" explicitly in the query. This difference by itself is not a shortcoming of either of the two models.

The identities for the $e\downarrow\phi$ operator in the parametric model stated above are a direct counterpart of those in the WSQ model. In particular, observe that in the parametric model the identity $(e_1 - e_2) \downarrow \mu = (e_1 \downarrow \mu) - (e_2 \downarrow \mu)$ holds. Thus the difference operator in the parametric model is well behaved. We note that the $e\downarrow\phi$ operator in the parametric model works cleanly in every conceivable context. Consider the following remarks about the level shift operator $e \uparrow U$ in the WSQ model.

- It is stated in [WSQ94] that because of the unary nature of relation U in the level shift operator $e \uparrow U$, U cannot be involved in a projection, a selection or a cartesian product. In the parametric model these possibilities do not arise because the domain expressions are not relations, they are simply time domains. In addition, the syntax they lead to is simple, powerful, and uniform.
- It is stated in [WSQ94] that U cannot involve the difference operator. In the parametric model no such problem arises: $e \downarrow (\mu_1 - \mu_2)$ is allowed, and the natural identity $e \downarrow (\mu_1 - \mu_2) = e \downarrow \mu_1 - e \downarrow \mu_2$ holds.
- It is also stated in [WSQ94] that a cascade of level shift operators does not give rise to interesting identities in general. This is not a problem in the parametric model, where the natural identity $(e \downarrow \mu_1) \downarrow \mu_2 = e \downarrow (\mu_1 \cap \mu_2)$ holds.

7. Querying the multilevel security data

In the following we exhaustively cover all queries from [WSQ94][1]. Let's compare how these queries are expressed in the WSQ and the parametric models. We find that the parametric model, where the queries are usually simpler than those in the WSQ model has a distinct advantage. Recall that the constant *me* stands for the user who submits a query. We will now use the variable *Users* to denote the space of all user levels.

Example 9. *List my belief about John's department*. For this query, the expressions in the algebras of the WSQ and parametric models are given below. The expressions illustrate the difference in the defaults used in the two models. In the WSQ model the query is executed only on the data at the level where the query is submitted. On the other hand in the parametric model it would be executed on the whole database, necessitating explicit restriction to *me*.

WSQ model: $\Pi_{Dept}\,\sigma$(emp, Name = John)

Parametric model: $\Pi_{Dept}\,\sigma$ (emp, Name = John, *me*)

1. Note that all the queries in this section have appeared in [WSQ94]. They have been adapted to the emp relation, our running example.

The results of a level shift operator can be unexpected. As evidence, we observe that $(e_1 - e_2)\uparrow\mu$ can be a proper superset of $(e_1\uparrow\mu) - (e_2\uparrow\mu)$. This is shown in the following counter example.

Example 8. Consider the database scheme {r(A), s(A)}. Suppose that the instances of r(A) and s(B) at different levels are as follows:

u_3: {a,b} and ∅, respectively.[1]

u_2: {a,b} and {a}, respectively.

u_1: {a,b} and {b}, respectively.

Given the above, we have

$(r - s)\uparrow$anyone = $(\{a,b\} - \varnothing) \cup (\{a,b\} - \{a\}) \cup (\{a,b\} - \{b\}) = \{a,b\}$

$r\uparrow$anyone = $\{a,b\} \cup \{a,b\} \cup \{a,b\} = \{a,b\}$

$s\uparrow$anyone = $\varnothing \cup \{a\} \cup \{b\} = \{a,b\}$

$r\uparrow$anyone – $s\uparrow$anyone = ∅

Therefore, $(r-s)\uparrow$anyone is a proper superset of $r\uparrow$anyone – $s\uparrow$anyone. •

In contrast, in the parametric model the relational difference operator will always behave as expected. The reason for this is that in the parametric model the user level can never be separated from a value; thus, the distinction between a value such as 55K at two different user levels is not ignored by the system.

6.1. The restriction operator in the parametric model

Now it is time to introduce an operator that comes closest to the level shift operator of WSQ. Recall the 1-3-selection, $\sigma(e, , \phi)$, for the parametric model for multilevel security. Here ϕ is a domain expressions, and as explained above ϕ is very versatile: it consists of subqueries that are relational, domain, and boolean expressions. We will use the abbreviation $e\downarrow\phi$ for $\sigma(e, , \phi)$.[2] The operator $e\downarrow\phi$ is called the *restriction operator*. The following identities could be proved for the restriction operator:

- $\Pi_X(e) \downarrow \mu = \Pi_X(e \downarrow \mu)$
- $\sigma(e, \phi) \downarrow \mu = \sigma(e \downarrow \mu, \phi)$
- $(e_1 \cup e_2) \downarrow \mu = (e_1 \downarrow \mu) \cup (e_2 \downarrow \mu)$
- $(e_1 - e_2) \downarrow \mu = (e_1 \downarrow \mu) - (e_2 \downarrow \mu)$
- $e \downarrow (\mu_1 \cup \mu_2) = e \downarrow \mu_1 \cup e \downarrow \mu_2$

It is appropriate to think of $e\downarrow\phi$ in the parametric model as the counterpart of the level shift operator $e \uparrow U$. The reader should be cautioned that the two operators are duals of each other because of the way defaults work in the two models. In the para-

1. Strictly speaking {a,b} should be written as $\{\langle a\rangle,\langle b\rangle\}$.

2. In [Ga88] $e\downarrow\phi$ was written as $e\phi$. Note the use of the down arrow in $e\downarrow\phi$, as opposed to the up arrow in the level shift operator $e \uparrow U$. The arrow direction seem appropriate: the restriction operator $e\downarrow\phi$ removes information from e, whereas the level shift operator $e \uparrow U$ adds information to e.

• The above query disregards the source levels in the final result. If this information is desired, instead of the query "emp$\uparrow$ anyone", one can pose the query "(emp$\times$self)$\uparrow$ anyone", which gives rise to the relation shown in Figure 8 (b). Note that the information contained in this relation is the same as that in the relation of Figure 1 (b), which is the counterpart of a temporal relation with tuple label stamping. •

As seen above, there are two types of operators in the WSQ model: the classical operators and the level shift operator. The classical operators obviously satisfy the classical identities. For the level shift operator, [WSQ94] lists several identities; the following is a sampling:[1]

- $\Pi_X(e) \uparrow \mu = \Pi_X(e \uparrow \mu)$
- $\sigma(e, \phi) \uparrow \mu = \sigma(e \uparrow \mu, \phi)$
- $(e_1 \cup e_2) \uparrow \mu = (e_1 \uparrow \mu) \cup (e_2 \uparrow \mu)$
- $(e_1 - e_2) \uparrow \mu \supseteq (e_1 \uparrow \mu) - (e_2 \uparrow \mu)$
- $e \uparrow (\mu_1 \cup \mu_2) = e \uparrow \mu_1 \cup e \uparrow \mu_2$

6. Discussion

This section includes some general remarks about the WSQ model. Recall the expression U participating in the level shift operator e$\uparrow$ U. Corresponding to the expression U in the WSQ model, the parametric model has domain expressions. These domain expressions are composed from the primitives Dom(u); the visibility domain of a user, $[\![A]\!]$, $[\![A\theta B]\!]$, $[\![A\theta b]\!]$, $[\![e]\!]$; and operators $\cup$, $\cap$, and $-$. Conceptually, the domain expressions are conceptually very simple: they evaluate to user domains. Note that in particular that e, in the primitive $[\![e]\!]$, is an arbitrary relational expression. Therefore the domain expressions are recursively composed from other relational, domain, and boolean expressions. The syntax associated with the domain expressions is also very simple when compared to expressions such as U in the WSQ model. (Several examples will be given in the next section.)

In the WSQ model there is a lack of uniformity between the stored and computed relations. For a given database scheme, there is an instance of that database scheme at each user level in the stored database. On the other hand, the computed relations are not placed anywhere in the user hierarchy. Even if the relation computed by e$\uparrow$ U was placed at the level of the user posing the query, a corresponding instance would not exist at other levels. This tends to make the level shift operator a terminal operator: that is, once it is applied, it cannot be applied again. It is difficult to use a computed relation and a stored relation as subqueries in a larger query. In contrast, in the parametric model the relational scheme is the same as the one in classical databases, and for each relation scheme in the database scheme there is only one relation in the database. No expression in the parametric model is terminal in the sense that it can be used as a subquery of a larger query.

1. A full discussion of the cross product for the parametric model would require a considerable machinery and it is omitted from this paper. Therefore, we have not listed an identity involving the cross product in [WSQ94].

Name	Salary	Dept
John	80K	Toys
Tom	60K	Shoes
John	50K	Toys

(a) The relation computed by B[anyone]emp

Name	Salary	Dept	User
John	Toys	50K	u_1
John	Toys	80K	u_2
Tom	Shoes	60K	u_2
Tom	Shoes	60K	u_3

(b) The relation computed by B[anyone]emp

Figure 8. Examples of level shift operator in the WSQ model

the other hand in the WSQ model when a user u poses a query to the system, the system executes the query only on the instance of the database available to user u. Thus a user can perform classical operators on the relations owned by him/her.

5.2. The level shift operator

In addition to the classical operators, the WSQ model introduces an operator, called the level shift operator. To ease the formalism associated with the level shift operator, let's first introduce a notation: if e is a relational expression, and u is a user level, then $e\uparrow u$ denotes the relation computed by the expression e with the data available only at level u. The *level shift* operator is of the form $e\uparrow U$, where e is any relational expression and U is a single column relational expression containing tuples that are user levels.[1]

$$e\uparrow U = \cup_{\langle u\rangle \in U} (e\uparrow u)$$

First e is evaluated at every level in U, and then the relations thus obtained are unioned together.

Example 7. Let's consider a few examples to illustrate the use of the self and anyone relations and the level shift operator B. In all these examples let's assume that the queries are being posed by the user u_3.

- The query "emp" returns the state of the emp relation at user level u_3.
- The query "$emp\uparrow \{\langle u_2\rangle\}$" returns the state of the emp relation at user level u_2.
- The query "$emp\uparrow$ anyone" is more interesting. To understand it, note that anyone at level u_3 is the relation $\{\langle u_3\rangle, \langle u_2\rangle, \langle u_1\rangle\}$. Therefore, the query computes the (ordinary) union $(emp\uparrow u_3) \cup (emp\uparrow u_2) \cup (emp\uparrow u_1)$. For the given state of the database, this computation leads to the relation shown in Figure 8 (a).

1. To be exact, U is a relational expression which should evaluate to a single column relation containing tuples that are user levels.

5. The WSQ model for multilevel security [1]

In the WSQ model, a universe $\{u_1, u_2, \cdots, u_n\}$ of users together with $\leq$ is postulated. For a given database scheme, each user in the hierarchy has his/her own level's instance of the database. In addition, each user also owns two relations: "self" and "anyone", each a single column relation over an attribute called "Label". The self relation for user u consists of the single tuple $\langle u \rangle$. On the other hand, the anyone relation contains a tuple for the user u and each user below u.

Example 6. Recall that in the running example, the database scheme consists of a single relational scheme emp with attributes Name Salary Dept. We have also postulated the universe $\{u_1,u_2,u_3\}$ of users, where $u_1 \leq u_2$ and $u_2 \leq u_3$. As shown in Figure 5, in the parametric model there is a single relation for all users, and every user has access to a portion of that relation. Figure 7 shows the multilevel security database in the WSQ model that corresponds to the running example. The database in the WSQ model contains nine relations the our running example. •

emp

Name	Salary	Dept
Tom	60K	Shoes

self

Label
u_3

anyone

Label
u_3
u_2
u_1

(a) The database fragment at level u_3

emp

Name	Salary	Dept
John	80K	Toys
Tom	60K	Shoes

self

Label
u_2

anyone

Label
u_2
u_1

(b) The database fragment at level u_2

emp

Name	Salary	Dept
Tom	60K	Shoes

self

Label
u_1

anyone

Label
u_1

(c) The database fragment at level u_1

Figure 7. A database in the WSQ model corresponding to Figure 5.

5.1. Classical relational operators

In the parametric model, when a user u poses a query, the system filters the emp relation to Dom(u), the domain visible to u. Thus by default the system is set to query information belonging to u as well as information belonging to users lower than u. On

1. For ease of reading, we make the following notational changes in syntax: the projection e[X] will be denoted as $\Pi_X(e)$, the selection e[ϕ] will be denoted as $\sigma(e, \phi)$, and the level shift operator $B[e_2]e_1$ (explained later in this section) will be denoted as $e_1 \uparrow e_2$.

user domains. In such a case, the user u is enrolled at an existing user level and no new user level is created. The alternative would be to choose Dom(u) as a union of some of the existing user domains. This is simply a way of saying that the new user is enrolled at a level that is immediately above the users whose domains have been unioned. One more condition should be added to complete the requirements for the case of multilevel security: Dom(u) must contain the level assigned to u, allowing a user to access his/her own data.

In summary, whereas in a temporal database one has the freedom to enroll any number of users assigning them arbitrary domains, the corresponding assignment of user domains in multilevel security is more constrained. The fundamental requirement is that a user in multilevel security should be able to see his/her own updates made to the database. In this sense, multilevel security is a special case of the temporal case, and not the other way around.

An interesting feature of the parametric approach is that Dom(u) can be integrated in the algebra as a primitive for domain expressions to the set of existing primitives $[\![A]\!]$, $[\![A\theta B]\!]$, $[\![A\theta b]\!]$, and $[\![e]\!]$. More complicated domain expressions can be formed using $\cup$, $\cap$, and $-$. This integrates the concept of a user tightly and seamlessly into the model making the concept of user an object of complex queries.

From now onward the terms user and user level will be interchangeably, and no confusion should arise. A few additional primitives useful for querying the parametric model for multilevel security will be added.

- *me*. When a user u poses a query, the system interprets "me" as u.
- Below (u′). When a user u poses a query, Below (u) is interpreted as Dom(u′) – {u′}.
- Above (u′). When a user u poses a query, and u′ is visible to u, then Above (u′) is interpreted as Dom(u′) – {u′}.

In order to present a simple but intuitive example, assume that the relational algebra contains a relational expression of the form "r", where r is a relation in the stored database.

Example 5. To adapt the running example to multilevel security, we assume the set of users $\{u_1, u_2, u_3\}$ such that $u_1 \leq u_2$ and $u_2 \leq u_3$. Suppose the user u_2 wants to see the current state of the emp relation. To do this he/she executes the query "emp". The query retrieves the result shown in Figure 6. •

Name	Salary	Dept
$\{u_2\}$ John	$\{u_2\}$ 80K	$\{u_2\}$ Toys
$\{u_2\}$ Tom	$\{u_2\}$ 60K	$\{u_2\}$ Shoes

Figure 6. The result of the query "emp" posed by user u_2

information at least 10 years old, the analyzer has the last 5 years worth of information, and the classical user only sees the current information (as would be the case in a classical database). •

4. Parametric model for multilevel security

The parametric model for temporal data discussed in the previous sections can easily be adapted to multilevel security. The terms instant and temporal element are changed to *user level* and *user element*, respectively. Corresponding to the universe of time $\{t_1, t_2, \cdots, t_n\}$ in the temporal case is the universe of *user levels* $\{u_1, u_2, \cdots, u_n\}$ in multilevel security. The relation of Figure 1 (a) in the parametric model for multilevel security will be as shown in Figure 5.

Name	Salary	Dept
$\{u_1,u_2\}$ John	$\{u_1\}$ 50K $\{u_2\}$ 80K	$\{u_1\}$ Toys $\{u_2\}$ Toys
$\{u_2,u_3\}$ Tom	$\{u_2,u_3\}$ 60K	$\{u_2,u_3\}$ Shoes

Figure 5. A relation corresponding to Figure 1 (b) for multilevel security

Note that in the parametric model for temporal data we did not impose any order properties on the instants. Clearly, the parametric model and its query language do not depend upon the order properties. In other words, if an order is imposed on the instants, it does not change the underlying model. The parametric model is generic, that is, it mainly depends upon the set theoretic primitive $\subseteq$ on parametric elements (temporal elements and user elements).

4.1. The user hierarchy in multilevel security

The primitive $\subseteq$ on parametric elements leads to a user hierarchy introduced in the previous section. The user hierarchy gives different users access to different portions of the database. In the parametric model a users u_1 is below u_2 in the user hierarchy if and only if $Dom(u_1) \subseteq Dom(u_2)$, where $Dom(u_1)$ and $Dom(u_2)$ are the domains assigned by the system to the users u_1 and u_2.

In multilevel security one encounters a special (less general) case of the user hierarchy. The difference is that in multilevel security, the domains are more rigidly determined by the system. A partial order $\leq$ among the user levels is postulated and Dom(u) is defined as $\{u' : u' \leq u\}$. The following property holds in the user hierarchy:

Proposition 1. If u_1 and u_2 are user levels, then $u_1 \leq u_2$ if and only if $Dom(u_1) \subseteq Dom(u_2)$.

Note that the primitive $Dom(\cdot)$ of parametric databases can be used to induce a partial order $\leq$ in a multilevel security. To understand this, suppose we choose to use $Dom(\cdot)$ as the primitive. When a new user u enrolls to use the database, the Dom(u) must be determined for that user. One choice is to let Dom(u) be one of the existing

select X
from r
restricted to ϕ
where f

A precise semantics of this form of select statement can be given easily in the parametric model in terms of selection and projection operators: $\Pi_X \sigma(r, f, \phi)$. As in the definition of the selection operator, the "restricted to" clause limits the retrieval of a tuple τ to the temporal element computed by $\phi(t)$. Several examples of the select statement will follow later in the paper.

3. Concept of user hierarchy in the parametric model

In the previous section, we implicitly assumed that there is only one user for the parametric model. Such a user has access to the whole history, i.e., values in the database during the entire time $\{t_1,t_2,\cdots,t_n\}$. To facilitate a clear comparison with the WSQ model, we must introduce the concept of a user hierarchy in the parametric model.

For the parametric model let's now hypothesize multiple users. Corresponding to every user u, we formally associate a temporal element in $\{t_1,t_2,\cdots,t_n\}$, called the *domain* of u, denoted as Dom (u). When a user u submits a query to a database, the system automatically restricts the database to Dom(u) before processing the query. Clearly, the set theoretic containment among users creates a partial order among users. Formally, we say that users u_1 is *below* u_2 in the user hierarchy if and only if $Dom(u_1) \subseteq Dom(u_2)$. A user hierarchy is shown in Figure 4.

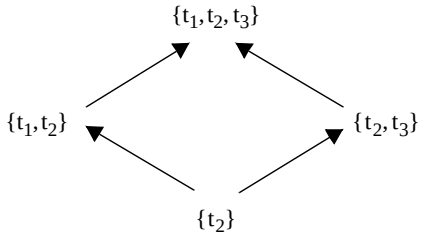

Figure 4. A user hierarchy for the parametric model

The concept of user hierarchy was introduced in [GB89]. There a useful and elaborate hierarchy has been given in a bitemporal model. The following covers an interesting example of users of the temporal model presented in the previous section.

Example 4. Imagine the temporal universe $\{1,2,\cdots,\text{NOW}\}$ also denoted as the interval [0,NOW], where NOW is the current instant of time. Now imagine that we have a database used by a governmental agency, which declassifies information after 10 units of time. Consider the following community of users: *system*, *public*, *analyzer*, and *classical* with user domains [0,NOW], [0,NOW−10], [NOW−4,NOW], and {NOW}, respectively. The system user can see the whole information, the public can only see

specify its snapshots and its key. In the following we assume that the key of r is K, and the natural join is denoted as $\Diamond$.

Operator	Definition of Snapshot	Designation of Key
Stored relation r	r(t)	Same as key of r
Union	$(e_1 \cup e_2)(t) = e_1(t) \cup e_2(t)$	Same as key of r and s
Difference	$(e_1 - e_2)(t) = e_1(t) - e_2(t)$	Same as key of r and s
Natural join	$(e_1 \Diamond e_2)(t) = e_1(t) \Diamond e_2(t)$	Union of keys of r and s
Projection	$(\Pi_X(e))(t) = \Pi_X(e)(t)$	If $K \subseteq X$ then K, else X
1-3-selection	$(\sigma(e, , \phi))(t) = \sigma(e(t), , \phi(t))$	Same as key of r, explained below

The definitions of union, difference, natural join ($\Diamond$), projection and 1-3-selection[1] given above are completely precise.[2] As an example consider the definition of union. $(e_1 \cup e_2)(t) = e_1(t) \cup e_2(t)$, which shows how snapshots can be computed. A snapshot of the union $(e_1 \cup e_2)(t)$ is defined as $e_1(t) \cup e_2(t)$. The latter is well defined as it is essentially a union of two classical relations. Thus, we have completely specified the snapshots as well as the key of $e_1 \cup e_2$. Therefore, $e_1 \cup e_2$ is well defined.

The 1-3-selection $\sigma(e, , \phi)$ needs more explanation. A 1-3-selection is a special case of selection of the form $\sigma(r, f, \phi)$, to be discussed below. In a 1-3-selection the second argument is left blank, and it is the operator in the temporal database that is a direct counterpart of the classical databases. In the 1-3-selection $\sigma(e, , \phi)$, the parameter ϕ is a domain expression. An example of the 1-3-selection is $\sigma(\text{emp}, , [\![\text{Dept} = \text{Toys}]\!] \cup [\![\text{Dept} = \text{Shoes}]\!])$. In temporal databases, it is a counterpart of the classical selection $\sigma(\text{emp}, \text{Dept} = \text{Toys} \vee \text{Dept} = \text{Shoes})$.

The general form of a selection is $\sigma(r, f, \phi)$. It evaluates to $\{\tau\downarrow\phi(\tau): \tau \in r, f(\tau)$ and $\tau\downarrow\phi(\tau)$ is not empty$\}$. If f evaluates to TRUE for a tuple, σ allows us to select only a relevant part of it, which is specified by ϕ. The key of $\sigma(r, f, \phi)$ is the same as the key of r.[3]

Example 3. The query *give information about employees while they were in Toys or Shoes if they are currently employed* can be expressed as follows:

$\sigma(\text{emp}, \text{NOW} \subseteq [\![\text{Name}]\!], [\![\text{Dept=Toys}]\!] \cup [\![\text{Dept=Shoes}]\!])$ •

The parametric model also includes an operator that allows a user to change the key of a relation [Ga88]. This operator has very interesting interaction with the selection operator [GN93]. In this paper we will also use an SQL-like select statement for our model. It turns out that for a comparison with the WSQ model, only a simple form of the select statement where the from list consists of a single relation will be needed. In other words, the select statement to be used in this paper is of the form given below:

1. The use of the term 1-3-selection is confined to this paper to make its relationship with [WSQ94] clearer. It is not a new operator in the parametric model.

2. Note that the snapshot semantics of the relational operators given here is a theoretical one. A more pragmatic semantics of the relational operators can be given directly without invoking the snapshots. This point is dealt with in more detail in [Ga86].

3. The definition of the full form of selection cannot be given in terms of snapshots [Ga88].

formed using temporal elements (e.g., $\{11,20\} \cup \{31,40\}$), $[\![A]\!]$, $[\![A\theta B]\!]$, $[\![A\theta b]\!]$, $[\![e]\!]$, $\cup$, $\cap$, complementation (unary $-$), and (binary) $-$, where A and B are attributes, b is a constant and e is a relational expression. If μ is a domain expression and τ is a tuple, then $\mu(\tau)$, resulting from the substitution of τ in μ, is a temporal element, and such substitution can be defined in a natural way. Following is an example of tuple substitution.

Example 2. Consider the domain expression $[\![\text{Salary} = 80K]\!]$. For a given tuple this expression retrieves the time domain where salary is 80K. Suppose τ is John's tuple in Figure 2. Then $[\![\text{Salary} = 80K]\!](\tau)$ evaluates to $\{t_2\}$. As another example, consider the domain expression $[\![\text{Salary} = 80K]\!] \cap -([\![\text{Dept} = \text{Toys}]\!] \cup [\![\text{Dept} = \text{Shoes}]\!])$. For a given employee, this expression retrieves the time domain consisting of instants where salary is 80K and the department is other than Toys or Shoes. For John's tuple, it evaluates to the empty set $\varnothing$. •

2.3. Boolean expressions

Boolean expressions are syntactic counterparts of boolean values TRUE and FALSE. They are formed using $\mu \subseteq \nu$, where μ and ν are domain expressions. More complex expressions are formed using $\wedge$, $\vee$, and $\neg$. Note that expressions of the form $\mu = \nu$, $\mu \neq \nu$, etc., can be derived using the above constructs. If t is an instant of time, $\{t\} \subseteq \nu$ can be written as $t \in \nu$.

2.4. Parametric syntactic forms $[\![A\theta B]\!]$ and $A\theta B$

We have already introduced the syntactic form $[\![A\theta B]\!]$ for the parametric model. In the parametric model the syntactic form $A\theta B$, without the use of $[\![\cdot]\!]$, is given a different meaning: $A\theta B$ is defined to be an abbreviation for the boolean expression of the form $-([\![A\theta B]\!] \subseteq \varnothing)$, which simply says that there is at least one instant of time where A is in θ relationship with B. Note that whereas $[\![A\theta B]\!]$ is a domain expression evaluating to a temporal element, $A\theta B$ is a boolean expression evaluating to TRUE or FALSE. Some important remarks about the parametric syntactic forms $[\![A\theta B]\!]$ and $A\theta B$ are now in order.

- The counterpart of the classical syntactic form $A\theta B$ in the parametric model is the parametric syntactic form $[\![A\theta B]\!]$ and not the syntactic form $A\theta B$.
- One of the uses of the parametric syntactic form $A\theta B$ is to identify objects. For example, "Name = John" is TRUE only for the first tuple in Figure 3.
- In a snapshot at an instant t, the distinction between the parametric syntactic forms $[\![A\theta B]\!]$ and $A\theta B$ essentially disappears. This is formalized in [BG93]. Therefore, the syntax in the parametric model is a consistent extension of that in the classical model.

2.5. Relational expressions

Before introducing relational operators, the concept of weak equality among relations must be defined. Suppose r and s are relations over the same scheme. Then r and s are said to be *weakly equal* if r and s have the same snapshots, i.e., $r(t) = s(t)$ for all instants t. It is easy to show that if two weakly equal relations have the same key, then the relations are equal. In other words, to specify a relation uniquely it is enough to

key attributes. Sometimes, the key attributes will be underlined for emphasis. Figure 2 shows a database with a relation emp(Name Salary Dept) with Name as its key. The relation is a counterpart of the temporal relation of Figure 1(c) in the parametric model.

Now suppose r is a relation. The *domain* of r, denoted $[\![r]\!]$, is defined as the union of domains of all tuples in r, i.e. $[\![r]\!] = \cup_{\tau \in r}[\![\tau]\!]$. Clearly, the domain of a relation is a temporal element. The *restriction* of r to temporal element μ, denoted $r{\downarrow}\mu$, is defined in a natural manner. The *snapshot* of r at an instant t, denoted r(t), is defined to be $r{\downarrow}\{t\}$. $[\![emp]\!]$, the domain of the emp relation of Figure 2, is $\{t_1,t_2\}$. The snapshot of the emp relation at instant t_2 is shown in Figure 3. The timestamp is not shown in this figure. Because of the homogeneity assumption, the snapshot of a temporal relation is isomorphic to a classical relation without nulls. In the parametric model, a database can be viewed as a parametrization of classical relations. Note that neither [WSQ94] nor this paper considers nulls.[1]

Name	Salary	Dept
$\{t_1,t_2\}$ John	$\{t_1\}$ 50K $\{t_2\}$ 80K	$\{t_1,t_2\}$ Toys
$\{t_2,t_3\}$ Tom	$\{t_2,t_3\}$ 60K	$\{t_2,t_3\}$ Shoes

Figure 2. emp relation of Figure 1(c) in the parametric model

Name	Salary	Dept
John	80K	Toys
Tom	60K	Shoes

Figure 3. Snapshot of the emp relation at $t = t_2$

Now let's present an algebra for the homogeneous relations. Our algebra includes three types of expressions: *domain expressions*, which evaluate to temporal elements; *boolean expressions*, which evaluate to boolean values (TRUE or FALSE); and *relational expressions*, which evaluate to relations. These three types of expressions are mutually recursive.

2.2. Domain expressions

Domain expressions are the syntactic counterparts of temporal elements. They are

1. [WSQ94] states: "Note that our formal treatment does not allow null values, just as ordinary relational algebra omits consideration of nulls. Null values may be included in a formal treatment by formalizing them in one of the many standard manners ..." The same remarks apply to the parametric model where the homogeneity assumption yields the counterpart of classical relations without nulls.

Note that there are an unusually large number of footnotes. Footnotes are necessary to keep the main text as easy to read as possible. However, let it be emphasized that the footnotes are an important and integral part of this paper.

2. The parametric model for temporal databases

The parametric model consists of a data type for time called temporal elements, attribute values, associative navigation (AθB), tuples, and relations. Our relations require a key to be designated with them. Finally, an algebra for the model will be introduced. The style of presentation is influenced by the need to make a clear comparison to the WSQ model.

Let's assume that the universe of time consists of instants $\{t_1, t_2, \cdots, t_n\}$. A *temporal element* is defined to be a finite subset of T. Note that no order properties are assumed for the set T.[1]

A *temporal value* of an attribute A is defined to be a function from a temporal element into the domain of A. A temporal value is also called an *attribute value* or simply a *value*. An example of a temporal value of the attribute COLOR is $\langle\{t_1\}$ red, $\{t_2\}$ blue$\rangle$. $[\![A]\!]$ denotes the *domain* of a temporal value A. Thus $[\![\langle\{t_1\}$ red, $\{t_2\}$ blue$\rangle]\!] = \{t_1,t_2\}$. $A{\downarrow}\mu$ denotes the restriction of A to the temporal element μ.

Our counterpart of the construct AθB for the relational model is $[\![A\theta B]\!]$, which is defined to be {t: A and B are defined at t, and A(t)θB(t) is TRUE}, the set of instants where A is in θ relationship to B. $[\![A\theta B]\!]$ is a temporal element. For example, $[\![\langle\{t_1,t_3\}$ red, $\{t_2\}$ blue$\rangle = \langle\{t_1,t_2\}$ blue$\rangle]\!] = \{t_2\}$. We also allow the construct $[\![A\theta b]\!]$, where b is a constant, which is evaluated by identifying b with the value $\langle\{t_1, t_2, \cdots, t_n\}$ b$\rangle$.

A *homogeneous tuple* τ over a scheme R is a function from R such that for every attribute A in R, $\tau(A)$ is a temporal value of A and all the temporal values in the tuple have the same domain. Informally, we say that a tuple is a concatenation of temporal values whose temporal domains are the same. The assumption that all temporal values in a tuple have the same domain make our tuples *homogeneous*.

Suppose tuple τ is given. Then the temporal domain of τ is the temporal domain of any attribute and is denoted by $[\![\tau]\!]$. A tuple is said to be *void* if its domain is empty. If μ is a temporal element, $\tau{\downarrow}\mu$ is obtained by restricting each value in τ to the temporal element μ.

2.1. Relations

Every set of tuples over a scheme R is not considered to be a relation. A *relation* r over a scheme R, with $K \subseteq R$ as the *key* of r, is a finite set of non-void tuples such that no key attribute value in a tuple changes with time, and no two tuples match in all their

1. In general the universe of time and temporal elements can be more complex. For a clearer comparison with the WSQ model this simple definition suffices. The main property which encapsulate temporal elements is their closure under union, intersection, and complementation.

• U-polyinstantiation.[1] Under *u-polyinstantiation* it is assumed that a real world object has the same key under all beliefs, although the nonkey values may vary. For example, Name of an employee would be the same in all beliefs, but varying beliefs about salary and department may exist. Because an object value may be different at two instants of time or at two points in space, u-polyinstantiation is in a mathematical sense a priori present in any model of temporal or spatial databases. (See Figure 1(c).)

• Key-polyinstantiation. *Key-polyinstantiation* allows key as well as nonkey attributes value to vary across beliefs. Key-polyinstantiation subsumes u-polyinstantiation. The concept of a polykey, where an object may have several key values, was introduced in [BG89,GB89,BG90][2] for temporal beliefs, and a brief discussion can be found in [GN93]. A discussion of key-polyinstantiation is beyond the scope of this paper. The key-polyinstantiation in multilevel security has been covered in [CG95, CG96].[3]

U-polyinstantiation seems to be the only form of polyinstantiation in multilevel security literature, where it is typically supported by having multiple tuples for a real-world object (see Figure 1(b)). As stated above, in a mathematical sense u-polyinstantiation is a priori present in any model of temporal or spatial databases. In some non-1nf models, u-polyinstantiation is captured at the tuple level. In addition, in the parametric model u-polyinstantiation is used as a keying mechanism for tuples: there is a one-to-one correspondence between objects in the real world and the u-polyinstantiated tuples. A tuple, or the corresponding real world object, is identified by its *unikey*,[4] the unique key value. Because the parametric model captures u-polyinstantiation at the tuple level rather than at the relation level, it seems to provide a cleaner framework for multilevel security databases with u-polyinstantiation.

In this paper we will consider [WSQ94] as a case study, and for this purpose we term the model presented in there as the *WSQ model*. This paper will focus on a comparison between the parametric and the WSQ frameworks for modeling and query of multilevel security data. The rest of this paper is organized as follows. Section 2 gives a brief introduction to the parametric model for temporal data. A user hierarchy for the parametric model is presented in Section 3. Section 4 shows how to adapt the parametric model and the user hierarchy to multilevel security. Section 5 introduces the WSQ model for multilevel security. Section 6 examines some characteristics of the two models for multilevel security. Section 7 exhaustively covers all queries in [WSQ94] and shows that they can be expressed more naturally in the parametric model. The conclusions are presented in Section 8.

1. The prefix "u-" in "u-polyinstantiation" may be seen as an abbreviation of "uni", the term "uni-polyinstantiation" would sound odd, therefore, we have coined the term "u-polyinstantiation".

2. To the best of our knowledge, the concept of key-polyinstantiation was first introduced in [GB89], where no special term was used for it.

3. Belief data is covered extensively in our works available in a series of six Technical Reports of which [Ga97] serves as an index.

4. The term unikey is used in this paper for brevity and also to make a clear distinction from polykeys.

• On the day that the user at the upper level wants to leak the bit "0", he/she does nothing. On the day the user at the upper level wants to leak the bit "1", he/she inserts a fictitious record for John in the morning and deletes the record in the evening.

• Every afternoon, the user at the lower level tries to insert a (fictitious) record for John. On some days the insertion will go through, and on other days the system will reject the insertion as a violation of the key. If the insertion is confirmed by the system, the user at the lower level assumes that the bit "1" has been sent to him/her by the user at the upper level and the lower-level user deletes the record just inserted. If the system rejects the insertion, the user at the lower level assumes that the bit "0" has been sent by the user at the upper level. •

Name	Salary	Dept
John	50K	Toys
Tom	60K	Shoes

(a) A classical relation with Name as its key

Name	User	Salary	Dept
John	u_1	50K	Toys
John	u_2	80K	Toys
Tom	u_2	60K	Shoes
Tom	u_3	60K	Shoes

(b) A multilevel security relation with Name User as its key

Name	User	Salary	Dept
John	t_1	50K	Toys
John	t_2	80K	Toys
Tom	t_2	60K	Shoes
Tom	t_3	60K	Shoes

(c) A temporal relation with similar mathematical content as (b)

Figure 1. Polyinstantiation in multilevel security and temporal databases

Typically, in multilevel security literature the covert channel is avoided by adding a "User-level" column to a relation so that the key is not just the Name attribute, but rather "Name and User-level" attributes put together. (See Figure 1(b).) With this arrangement, the system does not give an error message about duplication of a record. For this solution, the term polyinstantiation has been coined: *polyinstantiation* means the ability of a system to accommodate multiple beliefs about a real world object in the database. It turns out that there are two levels of polyinstantiation.

Technical Report: TR97-17; Computer Science Department, Iowa State University

Applicability of temporal data models to query multilevel security databases: a case study

Shashi K. Gadia
Computer Science Department
Iowa State University
Ames, IA 50011

gadia@cs.iastate.edu

Abstract. In a multilevel security database there are multiple beliefs about a given real world object. The ability of a database model to accommodate multiple beliefs is termed polyinstantiation in the multilevel security literature. In this paper we remark that in an abstract sense polyinstantiation is a priori present in all models for temporal and spatial databases. In particular we investigate the applicability of the parametric model for temporal data to query multilevel security data and, as a case study, compare it to a model for multilevel security given by Winslett, Smith, and Qian.

Index terms. Databases, relational databases, multilevel security, belief data, polyinstantiation, temporal databases, spatial databases, dimensional databases.

1. Introduction

Several models for temporal data have been proposed for which [Ta+93] is an excellent reference. For the parametric model for temporal data that originated in [Ga88], [GY88] provided the concept of a key, and [BG90] provided an SQL-like language. A summary of the parametric model given in [GN93] has also appeared in [Ta+93]. A brief summary of the parametric model will also be given in this paper.

Models for multilevel security have appeared in [BL75, CS95, DLS87, DLS88, GQ95, HOT91, JS90, JS91, LDS90, SW92, WSQ94]. In multilevel security there is a hierarchy of users or user levels, in which every user level has its own version of information. A user can see all information belonging to users at and below his/her level. On the other hand, the information belonging to a higher user level, or even existence of such information or such user levels, is held confidential from the lower user levels. A model for a multilevel security database must be devoid of a sort of communication, called a *covert channel*, which can lead to a compromise of the user confidentiality. A simplistic use of the classical first normal form database model, where every value is atomic, is venerable to covert channels. This is shown in the following example.

Example 1. Imagine a classical emp relation as in Figure 1(a) with Name as its key. Postulate two user levels, upper and lower, and assume that John is known to be a fictitious person at both the user levels. Everyday, a user at the upper level can leak one bit, 0 or 1, of some secret message to a user at a lower level as follows: