数理解析研究所講究録
第461巻 1982年 278-287

データベースビューサポートシステムに必要なマン・マシン・インタラクションについて

東北大学 電気通信研究所
増永良文

1. Introductory Remarks

In the relational database systems, it tends to support views for giving users a database macro facility (STON75) and providing system designers with a tool of authorization and security (CHAM75). Usually, a view is defined as a virtual relation which is derived from the base relations, i.e. the stored relations by using certain view defining operations such as the relational algebra operations. Obviously, no difficult problem arises so long as the views are offered to users to retrieve data. However, when the users want to update them, i.e. to delete, insert and rewrite them, then the difficult problems arise and sometimes it is impossible to update them (CODD74). The reason is that a view is generally virtual, so that an update to it is only effected if and only if it is translatable into updates to the base relations, which translation must satisfy certain criteria discussed below. At present, because of the difficulty of the view update translation problem, very restricted type of views such as the views composed of not more than one stored relation, for example, is only supported for updates in a commercial system (IBM81).

Now, let R(A1, A2, ..., An) be a relation which is defined as a subset of the direct product dom(A1) x dom(A2) x ... x dom(An) of n attributes domains. By att(R), we denote a set of all attributes $\{A1, A2, \ldots, An\}$ of R. In our framework, a view is defined as follows: (1) A base relation is a view. (2) Let V be a view. Then the projection of V on $X(\subseteq att(V))$, denoted by V[X], and the θ-restriction of V on X, $Y(\subseteq att(V))$, denoted by $V[X\ \theta\ Y]$, are views which are called a projection view and a θ-restriction view respectively where θ denotes any of the relations

=, $\neq$, <, $\leq$, > and $\geq$. (3) Let V and W be views. Then the direct product V x W, the union V∪W, and the difference V - W are views which are called a direct product view, a union view and a difference view respectively. (4) A relation is called a view if and only if it is derived by the derivation rules (1), (2), and (3). For example, a view ERP(EMP, $ROOM_E$, PHONE) is derived from two base relations ER(EMP, $ROOM_E$) and RP($ROOM_P$, PHONE) as follows: ERP = ((ER x RP)[$ROOM_E$ = $ROOM_P$])[EMP, $ROOM_E$, PHONE], where E and P are subscipted to ROOM to distinguish that $ROOM_E$ and $ROOM_P$ are ROOMs which belong to ER and RP respectively. Figure 1 shows the instances, i.e. the present value of the relations, of the view ERP, the base relations ER and RP, and the intermediate views W_1 = ER x RP and W_2 = W_1[$ROOM_E$ = $ROOM_P$]. (The reader will be noticed that ERP is the natural join of ER and RP, which definition is given in our framework.)

2. View Update Translation Problem.

As noticed in the previous section, the updates to views are only effected if they are translatable into updates to the base relations in that the translation must satisfy certain correctness criteria in order to avoid update anomalies. Our criteria are as follows (MASU82): (a) No extra update happens in the view by the translation. (b) No extraneous update happens in the base relations by the translation. (c) No semantic ambiguity happens in the translation process. That is, the criterion (a) prescribes that, for example, the insertion of only one tuple (e5, r1, p1) to the view ERP should not be accepted because otherwise an extra tuple (e5, r1, p2) will be inserted. The criterion (b) prescribes that, for example, the deletion of a set of tuples {(e1, r1, p1), (e1, r1, p2)} from ERP should be translated into the deletion of a tuple (e1, r1) from ER and should not be translated into any others. The criterion (c) restrains the occurrence of the semantic anomaly in the updated database. For example, the deletion of a set of tuples D_0 = {(e1, r1, p1), (e1, r1, p2), (e2, r1, p1), (e2, r1, p2)} from ERP is translatable into one of the following three deletions each of which satisties the criteria (a) and (b): (i) The deletion of a set of tuples {(e1, r1), (e2, r1)} from ER, (ii) the deletion of a set of tuples {(r1, p1), (r1, p2)} from RP, or (iii) the both deletions. Notice, however, that the semantics of those three deletions differ each other and therefore the updated database becomes to have different semantics. (That is, (i), (ii) and (iii) correspond to the facts that the deletion is issued because (i) the employees e1 and e2 have left the room r1, (ii) the phones p1 and p2 have

been removed from the room r1, and (iii) both happened respectively.)

Now, we investigate how the translation of an update to a generally defined view can be done which satisfies the above three criteria. First, Table 1 summarizes the translatability of updates to five basic views which are the direct product view, the union view, the difference view, the projection view, and the θ-restriction view (MASU82). It should be mentioned that (1) the semantic ambiguity problem$_1$ (SAP$_1$ for short) should be solved when the translation of a deletion to a difference view is tried, (2) the semantic ambiguity problem$_2$ (SAP$_2$ for short) should be solved when the translation of an insertion to a union view is tried, and the unique extrapolation problem (UEP for short) should be solved when the translation of an insertion to a projection view is tried. These problems essentially need man-machine interaction to be solved, and it is discussed in more detail in the next section. However, before going to the next section, we must overview our approach to built an entire view update translation mechanism: First, we recognize that the translation rules given in Table 1 are "local" in the sense that they dominate only an update translation from a node into its direct descendant node(s). (The reader can easily see that a generally defined view is represented as a tree, we call it a view defining tree, with the view at the root and the base relations necessary to define it at leaves.) Now, if all the local translations of a given view update are successful, then the view update is obviously realizable. But, notice that this translation scheme gives only a sufficient condition for the translatability of an update to a generally defined view (MASU80a, 82). In order to show a necessary and sufficient condition for it, we need the update modification rule which allows to modify a presently not translatable update to an intermediate view, i.e. a relation which is neither a view nor a leaf, so that it is translatable by a local rule, with the restriction that the intermediate view should be used to define a θ-restriction view as its direct ancestor. In order to deepen our understanding of this rule, let us think about the case in which the deletion D_0 (this is defined previously) is issued to the view ERP. First, it is easy to see that D_0 is translatable into the deletion D_1 to W_2 where D_1 = {(e1, r1, r1, p1), (e1, r1, r1, p2), (e2, r1, r1, p1), (e2, r1, r1, p2)} by Rule D-4. Second, D_1 becomes as it is the deletion to W_1, which we name D_2 anew, by Rule D-5. Third, we now recognize that the local translation of D_2 is impossible because the cross reference condition, which is a syntactic condition stating that $W_1 - D_2$ should be again a direct product, does not hold

for $W_1 - D_2$. The point is that it is hasty to judge that the translation of D_0 becomes impossible. That is, because W_1 is an intermediate view and only a subset of it, i.e. a set of tuples each of which has the equal $ROOM_E$ and $ROOM_P$ values, constitutes the direct ancestor W_2, it is possible to modify D_2 to D_3 such that (a) $D_3 \supset D_2$, (b) $D_3 \cap W_2 = D_2$, and (c) $W_1 - D_3$ satisfies the cross reference condition. For example, let us modify D_2 to $D_3 = D_2 \cup \{$(e1, r1, r2, p3), (e1, r1, r2, p4), (e2, r1, r2, p3), (e2, r1, r2, p4)$\}$. Then D_3 becomes translatable into the deletion $D_4 = \{$(e1, r1), (e2, r1)$\}$ to the base relation ER by Rule D-1. It is easy to see that this translation can realize the intended deletion D_0 to ERP. We call such a rule the deletion modification rule.(The insertion modification rule can also be definable.)

Now it happens a problem which we call the semantic ambiguity problem$_3$ (SAP_3 for short). That is, although we modified D_2 to D_3, there are yet other ways to modify it. For example, we can modify it to $D_3' = D_2 \cup \{$(e3, r2, r1, p1), (e3, r2, r1, p2), (e4, r2, r1, p1), (e4, r2, r1, p2)$\}$ or ever $D_3'' = D_3 \cup D_3'$. Obviousy, D_3' is translatable into the deletion $D_4' = \{$(r1, p1), (r1, p2)$\}$ to RP and D_3'' is translatable into both D_4 and D_4'. Other modifications are not possible due to the translation criterion (b). Now, which one of the three modification alternatives should be chosen? This is important because these have completely different semantics in the sense that modifying D_2 to either D_3 or D_3' or D_3'' must correspond to the following three facts, i.e. (i) the employees e1 and e2 have left the room r1, (ii) the phones p1 and p2 have been removed from the room r1,, or (iii) both happened respectively. In order to satisfy the translation criterion (c), we need to dissolve this ambiguity, i.e. to solve SAP_3. (Notice that the translation ambiguity of D_0 investigated at the beginning of this section comes from this ambiguity.)

3. Solvability of Four Problems with Relation to Man-Machine Interaction.

As it was mentioned before, the translatability of a view update completely depends on the solvability of SAP_1, SAP_2, SAP_3 and UEP which essentially require man-machine interactions in certain case. We will examine each of the four problems in turn and consolidate the basis of designing a view update translator involving man-machine interaction.

First, SAP_1 occurres in translating a deletion D issued to a difference view $V = R - S$. (R and S are union-compatible relations (CODD72).) That is, a deletion of a tuple t from V could be realized either by (i) deleting t from R, or (ii) inser-

ting t to S, or (iii) doing both (i) and (ii). Of course, the arbitrary choice should not be done because of the translation criterion (c). In some case, we can dissolve this ambiguity using extra semantic information. For example, when R is a relation of all employees of a company and S is a relation of all female employees of the company, then, by taking into account the fact that a man is either male or female, we can derive that every tuple deletion to R - S should always be translated into the deletion to R. However, let us think about another case in which R is defined as a set of all employees owning a Toyota car, and S is defined as a set of all employees owning a Datsun car. Then, when a deletion of an employee e from R - S is issued, we must identify the fact existing behind the deletion in order to dissolve the translation ambiguity, which is either (i) e has sold all e's Toyota cars without purchasing any Datsun car, or (ii) e has purchased a Datsun car without selling any e's Toyota car, or (iii) e has purchased a Datsun car with selling all e's Toyota car. Now the reader will see it easily that this soft of ambiguity may not be dissolved without involving man-machine interaction.

Second, SAP_2 occurs in translating an insertion to a union view. That is, let us define R and S as the relations of Japanese novels and American novels respectively, and suppose that the insertion of a book entitled "The house of the seven gables" is issued to RUS. Then, it is clear that we can not realize the insertion without identifying either the novel is a Japanese one or an American one, which identification will need man-machine interaction.

Third, SAP_3 occurs when we modify an intermediate update to a certain intermediate view. This problem has already mentioned in certain details in the previous section, and therefore ommitted here.

Last, UEP occurs in translating an insertion to a projection view. That is, in order to realize the insertion of a tuple t to a projection view R[X], where $X \subseteq att(R)$, we must find out a unique set of tuples $U = \{u \in dom(R)\}$ such that for every u in U, u[X] = t and the insertion of U to R is possible. (The number of such U must be at most one because otherwise the translation criterion (c) will be violated.) In certain cases, UEP can be solved easily. For example, in our translation example shown in the previous section, the insertion of (e, r, p) to W_2 is straightforwardly translatable into the insertion of (e, r, r, p) to W_1 by using the semantic information which comes from the syntactic nature of W_1 such that every tuple of W_1 has the identical $ROOM_E$ value with $ROOM_P$ value. However, in general, if we want to solve UEP, then it is not difficult to see that man-machine interaction will again be necessary, and may be it involves the null value

issue of the relational database which is not yet solved completely.

4. View Update Translator Design.

Now, Figure 2 illustrates the organization of the relational view update translator, which we propose. A complementary explanation is given below: (a) The system is hierarchically structured. Users sit at the top of the system and I_1 is the system manager by whom user requests to define views, or to update them, or the user answers to the questions issued by the view update translation manager will be translated into either the view manager or the view update translation manager. I_2 and I_4 are the front ends of the syntactic information base manager and I_3 and I_5 are the front ends of the semantic information base manager. (b) The view manager involves the view generator and the syntactic to semantic view information manager. The former analyzes the user definition of a view and generates the instance of the view and the view defining tree. The syntactic to semantic view information translator analyzes the view defining tree and extracts a certain semantic information. (An example of such information is that the $ROOM_E$ value is always identical with the $ROOM_P$ value for every tuple in W_2, which will be used by UEP later on.) (c) The view update translation manager involves the local view update translator and the global view update translator. In processing the local translation, SAP_1 solver, SAP_2 solver and UEP solver will be called, and in controlling the global translation, SAP_3 solver will be called. (d) In the syntactic information base, the information about the base relations and the local view update translation rules is stored. On the other hand, in the semantic information base, both the information which is extracted by the syntactic to semantic view information translator and the semantic information such as that the car type number 910 is a Datsun one, while 325 is a Mazda one, which will be obtained in the course of man-machine interaction, will be stored.

5. Concluding Remarks.

In this paper, we investigated the translatability of view updates in a relational database system with relation to man-machine interaction and illustrate the organization of the view update translator which inevitably involves the man-machine interaction. However, in order to make our design more concrete, we need further investigation of (i) the problem solving mechanism of the solver, (ii) the data structure of the semantic information base, and (iii) the specifi-

cation of languages such as the end user language, the system description language, and the data (or information) manipulating languages.

References

[CHAM75] Chamberlin, D. D., J.N.Gray and I.L.Traiger, "Views, Authorization, and Locking in a Relational Database System," Proc. AFIPS NCC, 1975, pp.425-430.

[CODD72] Codd, E.F., "Relational Completeness of Database Sub-languages," In Data Base Systems, Courant Computer Sci. Symp.6, R.Rustin, Ed., Prentice-Hall, Englewood Cliffs, 1972, pp.65-97.

[CODD74] Codd, E.F., "Recent Investigations in a Relational Database System," Information Processing 74, North-Holland Pub. Co., Amsterdam, 1974, pp.1017-1021.

[IBM81] SQL/Data System Concepts and Facilities, GH24-5013-0, File No.S370-50, IBM, January 1981.

[MASU79] Masunaga, Y., "On Data Processing Through User's Views in Relational Database Systems," (in Japanese) Paper of WGDBMS,WGDBMS 14-3, Information Processing Society of Japan, 1979.

[MASU80a] Masunaga, Y., "Design of a View Support Subsystem of a Relational DBMS using LUP," (in Japanese) Paper of WGDBMS, WGDBMS 19-1, Information Processing Society of Japan, 1980.

[MASU80b] Masunaga, Y. and S.Noguchi, "On Mappings between Schmata to Support User's Views in Database Systems," (in Japanese) Paper of TGAL, TGAL 80-23, Institute of Electronics and Communication Engineers of Japan, 1980.

[MASU81] Masunaga, Y., "Database Views as the Virtualization Method of Database," (in Japanese) Paper of WGDBMS, WGDBMS 24-2, Information Processing Society of Japan, 1981.

[MASU82] Masunaga, Y. and S. Noguchi, "A solution of the Database View Update Translation Problem based on a Semantic Approach," (in Japanese) submitted to the Transactions of Information Processing Society of Japan, 1982.

[STON75] Stonebraker, M., "Implementation of Integrity Constraints and Views by Query Modification," Proc. ACM SIGMOD Cof., 1975, pp.65-78.

ER:

EMP	$ROOM_E$
e1	r1
e2	r1
e3	r2
e4	r2

RP:

$ROOM_P$	PHONE
r1	p1
r1	p2
r2	p3
r2	p4

W_1:

EMP	$ROOM_E$	$ROOM_P$	PHONE
e1	r1	r1	p1
e1	r1	r1	p2
e1	r1	r2	p3
e1	r1	r2	p4
e2	r1	r1	p1
e2	r1	r1	p2
e2	r1	r2	p3
e2	r1	r2	p4
e3	r2	r1	p1
e3	r2	r1	p2
e3	r2	r2	p3
e3	r2	r2	p4
e4	r2	r1	p1
e4	r2	r1	p2
e4	r2	r2	p3
e4	r2	r2	p4

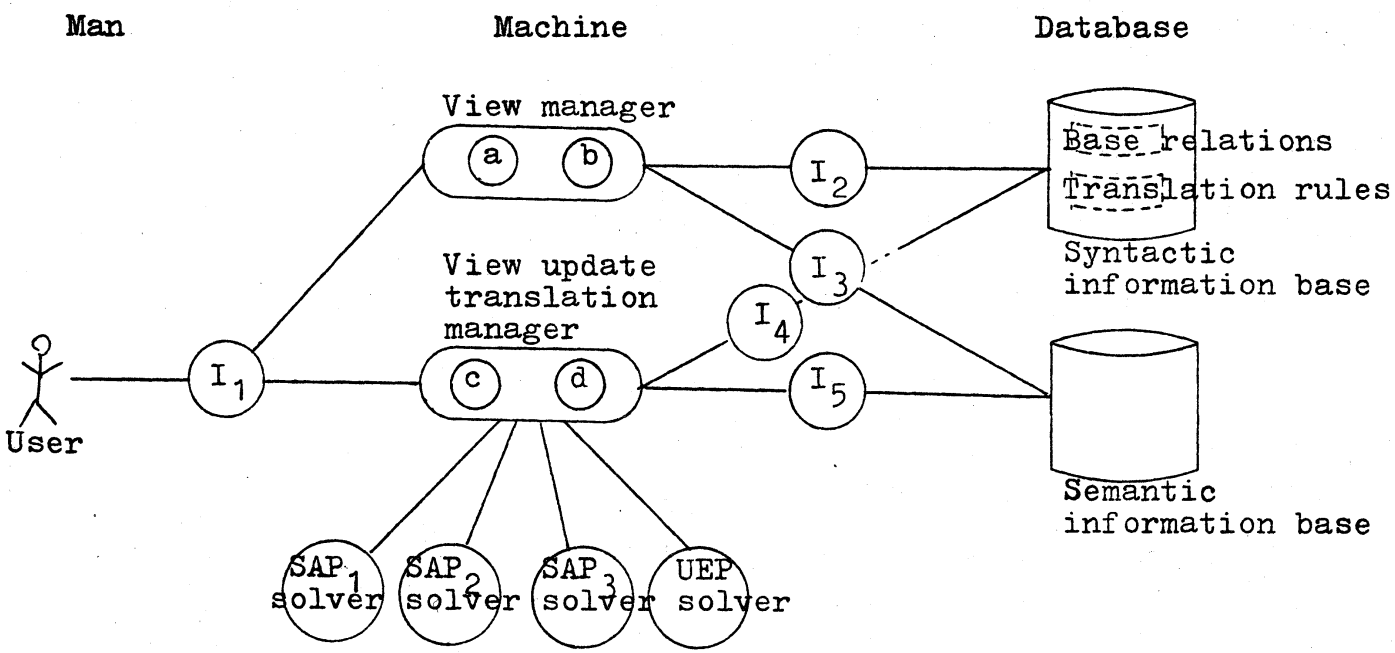
W_2:

EMP	$ROOM_E$	$ROOM_P$	PHONE
e1	r1	r1	p1
e1	r1	r1	p2
e2	r1	r1	p1
e2	r1	r1	p2
e3	r2	r2	p3
e3	r2	r2	p4
e4	r2	r2	p3
e4	r2	r2	p4

ERP:

EMP	$ROOM_E$	PHONE
e1	r1	p1
e1	r1	p2
e2	r1	p1
e2	r1	p2
e3	r2	p3
e3	r2	p4
e4	r2	p3
e4	r2	p4

Figure 1. Instances of ER, RP, W1, W2 and ERP.



a : View generator
b : Syntactic to semantic view information translator
c : Local view update translator
d : Global view update translator

Figure 2. The Organization of the View Update Translator.

Table 1. The Translation Rules of Deletion and Insertion to Five Basic Views.

(1) Deletion D.

Type of Basic Views	Translation Rule
$V = R \times S$	If V - D satisfies the cross reference condition, then translate D to the deletion (R - (V - D)[att(R)]) to R and the deletion (S - (V -D)[att(S)]) to S ...(D-1)
$V = R \cup S$	Translate D to the deletion D to R and the deletion D to S. ...(D-2)
$V = R - S$	If SAP_1 can be solved for every tuple in D, then translate D according to the result. ...(D-3)
$V = R[X]$	Translate D to the deletion $\{u \in R \mid \exists t \in D, u[X] = t\}$ to R. ...(D-4)
$V = R[X \theta Y]$	Translate D to the deletion to R as it is. ...(D-5)

(2) Insertion I.

Type of Basic Views	Translation Rule
$V = R \times S$	If V U I satisfies the cross reference condition, then translate I to the insertion (I[att(R)] - R) to R and the insertion (I[att(S)] - S) to S. ...(I-1)
$V = R \cup S$	If SAP_2 can be solved for every tuple in I, then translate I according to the result. ...(I-2)
$V = R - S$	Translate I to the insertion I to S and the deletion I to S. ...(I-3)
$V = R[X]$	If UEP can be solved, then translate I according to the result. ...(I-4)
$V = R[X \theta Y]$	If $t[X] \theta t[Y]$ holds for every tuple t of I, then translate I to the insertion to R as it is. ...(I-5)