Decisions and Decision Requirements for Data Warehouse Systems

Naveen Prakash[1], Deepika Prakash[2], Daya Gupta[3]

[1]MRCE, Sector-43, Faridabad, INDIA
praknav@hotmail.com
[2,3]Department of Computer Engineering, DTU, Delhi, INDIA
dpka.prakash@gmail.com, daya_gupta2005@yahoo.co.in

Abstract. We develop the notion of a decision requirement as the pair <decision, information> where 'information' is that required by the decision maker to assess if the 'decision' is to be taken or not. It is shown that there are two kinds of decisions, imperative and managerial. The former are decisions about which transactional service out of a choice of transactional services is to be provided. Managerial decisions determine what infrastructure out of a set of possibilities is to be put in place. It is shown that a decision is the reason why a functionality of an information system is invoked. The notion of decision requirement is clarified through a decisional requirement meta model. This is supported by a decision and information meta model.

Keywords: Decision, Information, Data Warehouse

1 Introduction

Goal oriented requirements engineering techniques [1-5] have been developed in the area of information systems/software engineering. These techniques aim to discover the functions of the system To-Be and lay the basis for system design.

The role of Requirements engineering in developing Data Warehouses has been investigated only in the last decade or so [6-13]. Today, there is a body of opinion that uses goal oriented techniques [10, 11, 13, 15, 16] for determining data warehouse structure. One goal-oriented approach [10, 11, 13] is based on the notion of the Goal-Decision-Information diagram. This approach postulates that the decision making capacity is determined by organizational goals. Additionally, it associates the information that has a bearing on a decision with the decision itself. In this paper, we represent this association as a pair, <decision, information> and refer to it as a **decision requirement**. Thus, in order to represent data warehouse contents, the set of decision requirements must be explicitly modeled.

Evidently there is a close relationship between the information systems and data warehouse of an organization. The former are used to populate the latter through the ETL process. In the opposite direction, the decision taken by using the data warehouse has the effect of changing information system contents. This means that information systems operate in a **decisional environment**. We consider this environment in the next section and show that there are two kinds of decisions, imperative and managerial. In the subsequent section we develop a meta model for decision requirements. Here we also model the notion of a decision and information from the data warehouse perspective. In section 4 we discuss our proposals with other related work.

2 The Decisional Environment

The decisional environment provides the context in which an information system (IS) operates. This is shown in Fig. 1. When the information system is sent a stimulus from the decisional environment then the functionality that responds to this stimulus is invoked.

Stimuli can be sent by two different kinds of actors, IS administrators and IS operators. These stimuli correspond to two kinds of decisions, managerial and imperative. Managerial decisions are used to 'initialize' the IS where as the latter work within the initialized IS to operate the system. For example, in a railway reservation system IS administrators initialize train data whereas IS operators invoke functionality to make reservations and cancellations using information set up by the IS administrator.

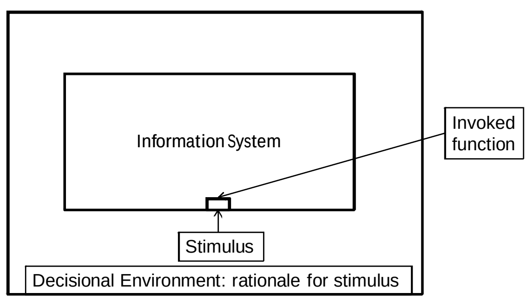

Fig. 1. Embedded IS in a Decisional Environment.

2.1 Imperative Decisions

Let there be a manager who has to perform extra work and needs to allot it to an employee. He can decide on the employee from the choice set {Transfer employee, Recruit employee, Overload employee}. The manager needs information to decide which alternative to pick and, also which individual employee shall be transferred, recruited, or overloaded respectively. There are two decision making problems here, to select from the choice set and to identify the individual, respectively. We shall use the notions of tactical decisions and operational decisions to classify these.

Fig. 2 shows the interplay of tactical and operational decisions. The tactical decision to Transfer an employee enters the operational decision making environment where the employee is identified and the stimulus to be sent to the information system is completely formulated. The information system performs the desired function and this information is now available to be sent to the DW at refresh time.

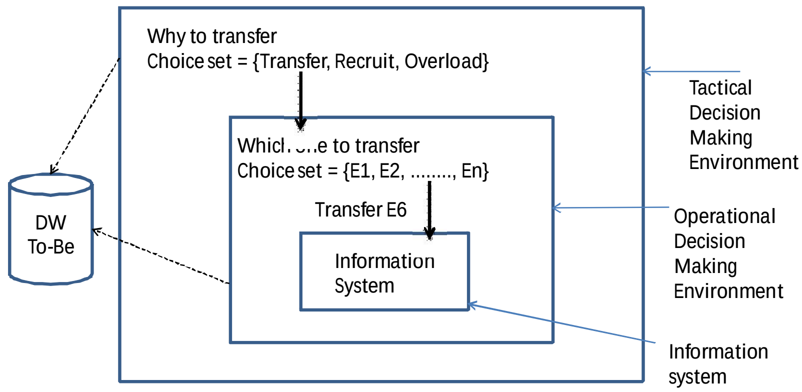

Fig. 2. Imperative Decisions and the interplay between tactical and operational decisions.

Looking from the information system outside, the decision making layers surrounding it formulate the stimulus to which the IS responds. This stimulus must identify IS functionality and the data. The former is done in the tactical environment whereas the latter is done in the operational decision making environment.

2.2 Managerial Decisions

There are two kinds of managerial decisions, those that follow a business policy, enforce it or create exceptions to it, and those that formulate the policy. We refer to the former as **administrative decisions**, since they are concerned with administering the system and to the latter as **policy decisions**. The latter provide the context for the former.

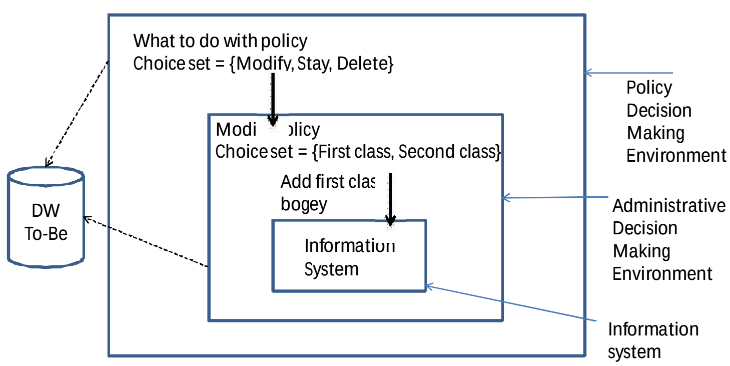

Fig. 3. Managerial Decisions

Let us be given a **policy decision** that the ratio of first class bogies in a train to second class bogies is 1:2. This policy is to be enforced as an **administrative decision**. **Policy decisions** may define the norms and standards that are used by administrative decisions or business rules used by imperative decisions. A policy decision requires knowledge of the state of the organization. For example deciding the 1:2 norm above requires the knowledge of patterns of bookings made, revenue targets, revenue receipts etc. Out of the many choices available to fix the ratio, the policy decision maker uses this knowledge to fix the desired one.

Fig. 3 shows that the policy decision to modify the ratio of first to second class bogeys in a train leads to the administrative decision to add a first class bogey, and the information system is stimulated to reflect the change. This information is now available for train reservation purposes and is also available to be sent to the DW.

3 Decision Requirement

We have seen that in order to make a decision reference to the information in the data warehouse needs to be made. We represent this as a pair <decision, information> and refer to it as a decision requirement. Here, we elaborate on the notion of decision requirement.

3.1 The Decision Requirement Meta-Model

The Decision Requirement, DR, meta-model is shown in Fig. 4. As shown it is modeled as an aggregate of information and decision.

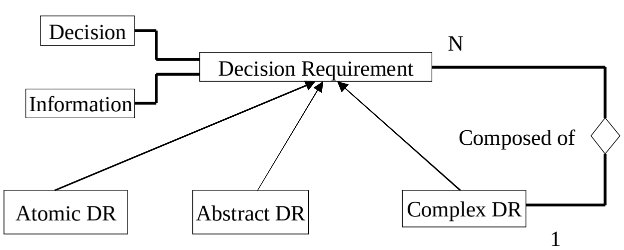

Fig. 4. Decision Requirements Meta-model.

Fig. 4 shows that there are three kinds of decision requirements, atomic, abstract and complex. An atomic DR is the smallest decision requirement. It cannot be decomposed into its parts.

An abstract DR is a decision requirement that is arrived using generalization/specialization principles. This gives rise to ISA relationships between decision requirements. Finally, a complex DR is composed of other simpler decision requirements. Complex decision requirements form an AND/OR hierarchy.

To illustrate an abstract DR, consider an automobile plant that makes 1-tonne and 13-tonne trucks. Let the decision of interest be *Set up New Assembly Line* and the required information be *Unsatisfied Orders.* This DR can be specialized into two DRs with decisions *Start New 1-tonne Line* and *Start New 13-tonne Line* respectively and required information, *Unsatisfied Orders for 1-tonners* and *Unsatisfied Orders for 13-tonners*.. Each of these is an ISA relationship with *Set up New Assembly Line*.

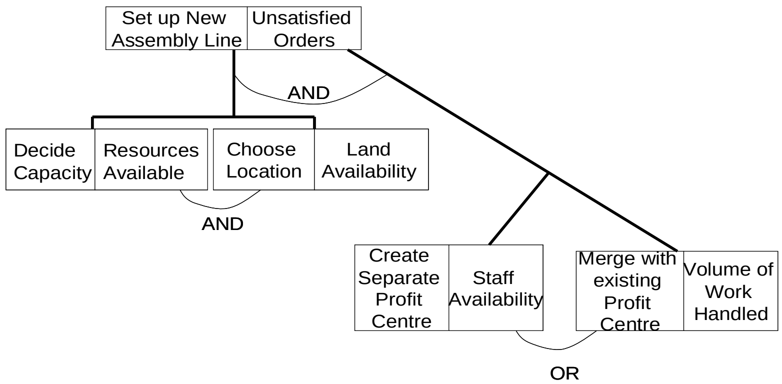

Fig. 5. Composition of Decision Requirements with AND and OR link

Now let us consider composition. The Decision Requirement <*Set up New Assembly Line, Unsatisfied Orders*> is a complex one having two component decision requirements, <*Decide Capacity, Resources Available*> and <*Choose Location, Land Availability*>. An AND link connects these two components so as to define the complex decision requirement, <*Set up New Assembly Line, Unsatisfied Orders*> (see Fig. 5).

The foregoing shows that a DR can be decomposed to reflect the decomposition of its decision component. It is also possible to do DR decomposition through information decomposition. In this case, the decision part is held constant whereas information components are elaborated. The Choose Location decision of Fig. 5 is shown as associated with the information, Land Availability. Land availability can be decomposed into two pieces of information, Land site and Land size Then the complex DR <Choose location, Land availability> can be decomposed into <Choose Location, Land site> and <Choose Location, Land size> respectively.

3.2 Meta-Model of Decisions

The key concept underlying the decision meta model of Fig. 6 is that of a decision parameter. Decision parameters reveal the factors that must be taken into consideration before a decision can be selected by the decision maker.

The decision to decision parameter relationship is M:N. A decision parameter must be associated with at least one decision. Similarly a decision must be associated with at least one decision parameter. **Dependent decision parameters** depend on other parameters for their existence whereas **independent decision parameters** determine a completely new aspect of a decision. Independent parameters may have dependent parameters but are themselves not dependent on any other decision parameter for their existence.

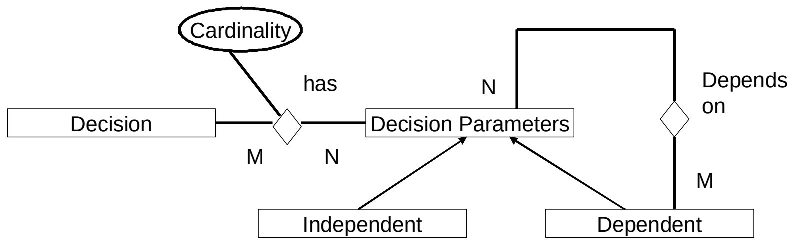

Fig. 6. Decision Meta Model

Consider the decision Set_Up_New_Assembly_Line(Product Type, Location, Line Capacity). Here, the parameters, Product Type and Location are independent of one another. In contrast, Line capacity is dependent on Product Type since it is determined by the type of the product built by the line.

3.2 Modeling Information

The information model in Fig. 7, shows three kinds of information, detailed, summarized or aggregates, and historical. Aggregate information is obtained as a summary by computing from simpler information. This is shown in Fig. 7, by the specialization of information into Simple and Aggregate as well as by the 'Is computed from' relationship between Aggregate and Information.

Historical information is represented by the relationship 'History of' between Information and Temporal unit. The cardinality of this relationship shows that it is possible for information to have no temporal unit associated with it. In such a case, only current information is to be maintained. However, when a temporal unit is associated with information then we must also know the number of years of history to be maintained. This is captured, as shown in the figure, by the attribute Period.

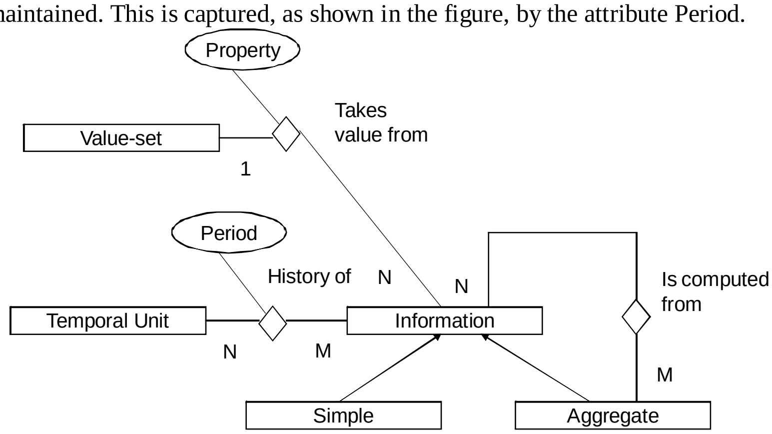

Fig. 7. Information Model in Data Warehouses showing three kinds of information.

Information is also associated with a value-set and takes on values from it. In Fig. 7 this association is called "Takes value from".

4 Comparison with Related Work

In traditional goal oriented requirements engineering, the aim is to specify system functionality. No support is provided in determining which of the many actions is to be performed. In our proposals, however, the focus is on the latter.

Our approach does not attempt to directly reach facts and dimensions unlike the database and ER driven approaches. Additionally, unlike these approaches, we can identify the required aggregate and historical information.

Goal oriented data warehouse development approaches of [6,7] and [16] reach data warehouse contents directly from goals without an explicit decisional stage. On the other hand, [15] recognizes the need to do further analysis from the decisional point of view. In contrast, we explicitly model the full decision making capability and associated information requirements.

Decision classification on the basis of time and planning horizon was proposed n GRAI grid [14]. The GRAI grid also provides an architecture of decisions of an organization. It provides a top level description of a system but does not aim to do requirements engineering for data warehousing.

Finally, our decisional environment is similar to the work system proposed in [17]. However, it addresses decision making, not operational information systems.

5 Conclusion

The notion of decision making implies the existence of a choice set from which the alternative that best meets organizational goals, is selected. These alternatives can be (a) managerial, for setting up the environment and (b) imperative, for providing the right service. Our emphasis is on modeling the set of decisions and associated information in an organization. It is only thereafter that one can proceed to subsequent stages of star schema design.

The ideas presented here have been tried out in a health scheme operating in India. Details can be obtained from the authors. Future work is centred round elicitation of imperative and managerial decisions.

References

1. Mylopoulos, J., Chung, L., Nixon, B., "Representing and Using Nonfunctional requirements: A Process-Oriented Approach", IEEE Trans. on Software. Engineering, Vol. 18 No. 6, June 1992, pp. 483-497.
2. Lamsweerde, A. van, "Goal-Oriented Requirements Engineering: A Guided Tour" Invited Paper for RE'01 - 5th IEEE International Symposium on Requirements Engineering, Toronto, August, 2001, pp. 249-263

3. Sutcliffe, A., Maiden, N., "Bridging the Requirements Gap: Policies, Goals and Domains", Proc. IWSSD-7 - 7th Intl. Workshop on Software Specification and Design, IEEE, 1993.
4. Yu E. S. K., "Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering", Proc. Third IEEE Symposium on Requirements Engineering, 226-235, 1997
5. Giunchiglia F., Mylopoulos J., Perini A., "The Tropos Software Development Methodology: Process, Models and Diagrams", Autonomous Agents and Multi-Agent Systems, 8, 3, 203-236, 2004
6. Boehnlein, M., Ulbrich vom Ende, A., Deriving initial Data Warehouse Structures from the Conceptual Data Models of the Underlying Operational Information Systems, Proc. Of Workshop on Data Warehousing and OLAP (DOPLAP), USA,15-21,1999
7. Boehnlein, M., Ulbrich vom Ende, A.: Business Process Oriented Development of Data Warehouse Structures. In: Proceedings of Data Warehousing 2000, Physica Verlag , 2000
8. Rilson F., Paim S., and Castro JFB, DWARF: An Approach for Requirements Definition and Management of Data Warehouse Systems, Proceeding of the 11th IEEE International Requirements Engineering Conference 1090-9, September 08 - 12, 2003.
9. Frendi, M., Salinesi, C.: Requirements Engineering for Data Warehousing, Proceedings of Workshop on REFSQ, 75-82, 2003
10. Prakash N., Gosain A., Requirements Driven Data Warehouse Development, CAiSE Short Paper Proceedings, 13-17, 2003
11. Prakash N, Singh Y, Gosain A., Informational Scenarios for Data Warehouse Requirements Specification, Conceptual Modeling-ER'2004, Atzeni P., Chu W., Zhou S., Ling T K(eds.) LNCS 3288, Springer, 205-216, 2004
12. Winter R., Strauch B., Information Requirements Engineering for Data Warehouse Systems. ACM Symposium on Applied Computing, Cyprus, 1359-1365, 2004
13. Prakash N., and Gosain A., An Approach to Engineering the Requirements of Data Warehouses, Requirements Engineering Journal, Springer, 13 (1), 49-72, 2008
14. Carrie AS and Macintosh R, An Assessment of GRAI Grids and their use in the Strathclyde Integration Method, production Planning and Control, 8, 2, 106-113, 1997
15. Golfarelli, M., Rizzi, S., Designing the Data Warehouse: Key Steps and Crucial Issues" in Journal of Computer Science and Information Management, Vol. 2. No.3, 1999.
16. Bonifati A.. Cattaneo F.., CeriS., A. Fuggetta, and S. Paraboschi. Designing Data Marts for Data Warehouses. ACM Trans. Softw. Eng. Methodol., 10(4):452–483, 2001.
17. Alter S, A General yet Useful Theory of Information Systems, Comm.of Association for Information Systems, 1, 13, 1-68, 1999