

# Decisions and Decision Requirements for Data Warehouse Systems

Naveen Prakash<sup>1</sup>, Deepika Prakash<sup>2</sup>, and Daya Gupta<sup>2</sup>

<sup>1</sup> MRCE, Sector-43,

Faridabad, India

praknav@hotmail.com

<sup>2</sup> DTU,

Delhi, India

dpka.prakash@gmail.com, daya\_gupta@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. The example of a health scheme is taken to illustrate the different kinds of decisions and decision requirements.

**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]. It has been argued [13] that Requirements engineering for DW systems focuses on **discovery of data** whereas for information/software systems, it focuses on **system functionality**. Early Data Warehouse development did not emphasize Requirements engineering. Indeed, Inmon [20] argued that requirements of data warehouses are usually the last thing to be discovered. Interest was in using existing databases for development of data warehouse systems. This was augmented by ER driven approaches where ER diagrams could be converted to star schemata. Now, 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. A given set of decision requirements evolves over time. The evolution of data warehouse is outside the scope of this paper. Further, this paper does not directly address the issue of conversion of decision requirements to data warehouse structures. We rely on the proposal of [13] to do this conversion.

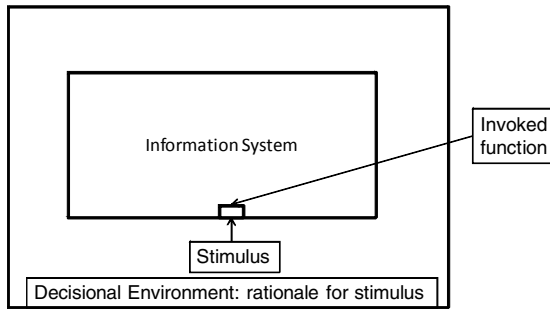
Evidently there is a close relationship between the information systems and data warehouse of an organization. The former are used to populate the latter through the ETL process. In the opposite direction, the decision taken by using the data warehouse has the effect of changing information system contents. 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 present a case study of a health scheme that uses the notion of a decision requirement. In section 5 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. The interface between the decisional environment and an information system is the stimulus sent from the former to the latter. When the information system is sent a stimulus from the decisional environment then the functionality that responds to this stimulus is invoked.

This implies that (see Fig.1) the decisional environment provides the rationale for the stimulus and a decision is taken here to send the appropriate stimulus out a set of possible alternative stimuli. For example, the information system may provide for transfer of an employee as well as for recruitment of a new employee. In the decisional environment, it is to be decided which of the two possible stimuli should be sent: depending upon salary budgets and whether another department can spare an employee or not, either a transfer stimulus or a recruitment stimulus will be sent. We can therefore say that the decision is the cause of the invoked functionality.

There are two different kinds of actors for an information system, IS administrators and IS operators. It is the job of the former to 'initialize' the IS. The latter then work within the initialized IS to operate the system. For example, consider a railway reservation system. IS administrators shall initialize the trains, their start stations, destinations and stop over stations, fares etc. that are used in operating the reservation system. They also handle changes: additions or deletions of trains, stations, changes in fare structure and so on. IS operators, in contrast invoke functionality to make reservations on given trains using the fare table and other information set up by the IS administrator.



**Fig. 1.** Embedded IS in a Decisional Environment

We see that there are two kinds of stimuli that the decisional environment can send: those decided by IS administrators and operators respectively. This leads to two kinds of decisions, that we call managerial and imperative decisions respectively.

## 2.1 Imperative Decisions

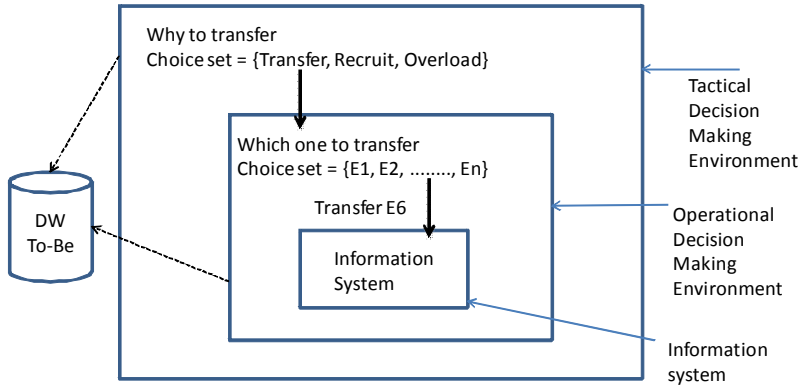
Let there be a manager who has to perform extra work and needs to allot it to an employee. He can find the employee by either transferring one from another department to this one, hire an altogether new employee or overload an existing employee with more responsibility. Thus, the choice set is {Transfer employee, Hire employee, Overload employee}. The manager needs information to decide which alternative to pick and, also which individual employee shall be transferred, hired or overloaded respectively. Thus, there are two decision making problems here, (a) to select from the choice set of functions as above, which we call tactical decisions, and (b) to identify the individual from the choice set of individuals that we refer to as operational decisions.

Tactical decisions supply the rationale for operational ones. In tactical decision making, the state of the organization is browsed through and analyzed to select from the choice set. For example, Transfer employee may be selected if a department is seen to be overstaffed, or an ill fitting employee has been posted there etc. Similarly, a new employee may be recruited because completely new skills are required.

The best alternative selected is the rationale behind an operational decision. It calls for formulation of the stimulus that shall be sent to the information system. Operational decision making is the process of selection of the appropriate individual from a choice set. For example, given the tactical decision Transfer employee, the operational decision maker needs to identify which one to transfer. The choice set consists of employees, and operational decision making leads to selection of one of these. Data about employees, their skills performance, past record, etc. is picked up from the data warehouse, analyzed and the most relevant employee data is sent as stimulus to the IS function that shall carry out the transfer.

Consider Fig. 2 which shows the interplay of tactical and operational decisions. The tactical decision making environment contains the operational decision making environment. Both environments refer to the data warehouse as shown by the dashed lines in the figure. Once a tactical decision is made, it is sent to the operational

decision making environment which sends a stimulus to the information system. Fig. 2 shows that 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

Since, the data warehouse is consulted in both forms of imperative decision making, full information about decisions and related information must be kept in it.

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

The class of managerial decisions is not concerned with operational functionality/transactional capability but with performing the administrative tasks to set up the system so that imperative decisions can be taken. Recall the reservation example cited earlier where it is necessary to initialize trains, fare tables etc.

We postulate that 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**. These two kinds of managerial decisions are related to one another. The latter provides the context for the former.

Let us be given a policy that the number of first class bogeys in a train should be one third of the train length. The rest are second class bogeys. In other words, the ratio first class: second class :: 1:2. This norm is usually to be followed. However, there could be exceptional situations like the train is making losses or it is overloaded.

In such situations this norm could be violated, and an administrative decision is needed to choose from the choice set, {Modify number of first class bogies, Modify number of second class bogies}. This administrative decision creates an exception to the norm.

Policy decisions are of two kinds. They define the norms and standards that are used by administrative decisions and also define business rules. They address the question of where a norm or standard comes from. Norms and standards may be defined externally, for example, by regulatory bodies, or internally within the organization. Assuming that external norms and standards are imposed on an organization, policy decision making is concerned with formulating internal ones. Consider the internal norm first: second:: 1:2 of the railway system considered earlier. Defining this through a policy decision requires knowledge of the state of the organization, 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.

The second type of policy decisions formulates business rules. Let it be that the number of days in advance a booking can be made is to be decided. Again, reference to existing information like pricing and booking pattern is made and the manager decides it to be 30 days.

The choice set available to the policy decision maker is {Define, Modify, Delete}.

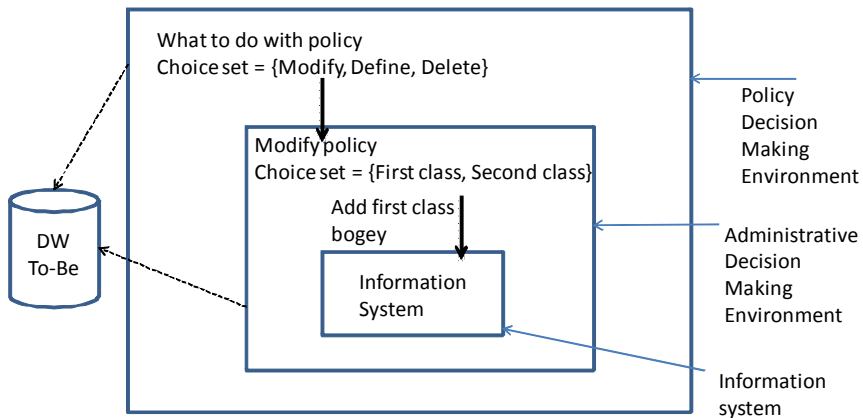


Fig. 3. Managerial Decisions

The interplay of policy and administrative decisions is shown in Fig. 3. Both the policy and administrative decision making environments refer to the data warehouse with the latter contained in the former. Once a policy decision is made, it is sent to the administrative decision making environment for enforcement. The information system is sent a stimulus from the administrative decision making environment to which it responds. Fig. 3 shows the policy decision to modify the ratio of first to second class bogeys in a train. The administrative decision to add a first class bogey is taken, 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.

Notice that even if a policy is held constant, an administrative decision may still be made. Thus, for example, even when the first class to second class ratio is not changed, an administrative decision to increase/decrease this ratio may be taken. This happens when there are sudden surges of demand for a particular kind of bogey.

As for imperative decisions, it is possible to support decision making of both kinds of managerial decisions. The data warehouse needs to be populated with information associated with these.

### 3 Decision Requirement

We have seen that in order to pick an alternative from the choice set, reference to the information in the data warehouse needs to be made. This information is a specification of the data to be kept in the data warehouse To-Be. We represent this association as a pair  $\langle \text{decision}, \text{information} \rangle$  and refer to it as a decision requirement.

In this section we elaborate on the notion of decision requirement. First we present the decision requirement meta model and then develop meta models for the notion of a decision and information.

#### 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. It can be expressed as a pair,

$$DR = \langle D, I \rangle$$

where D is a decision and I is the information required for it.

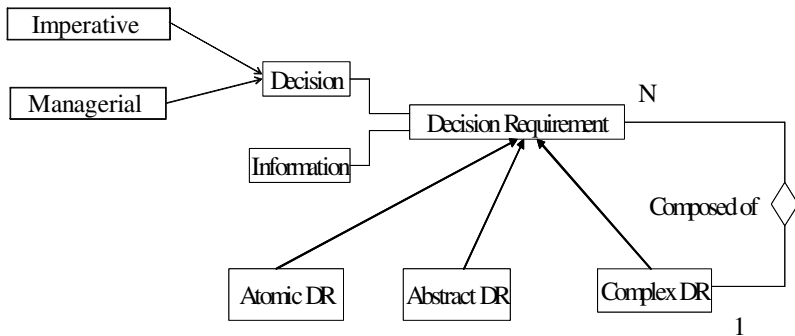
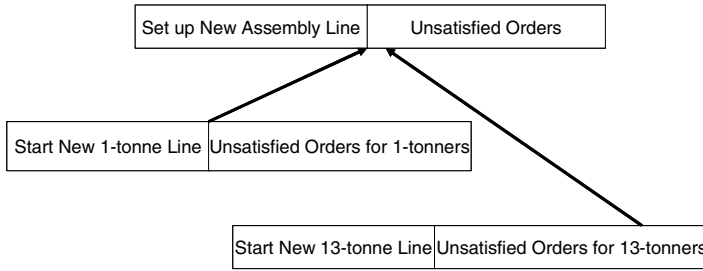


Fig. 4. Decision Requirements Meta-model

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.

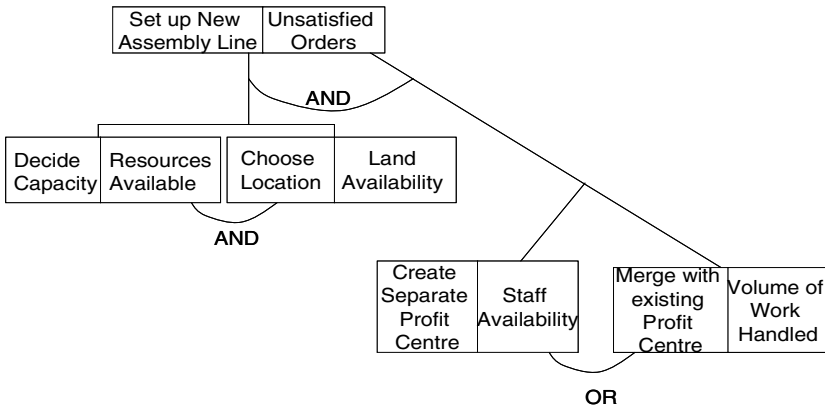
An abstract DR is a decision requirement that is arrived using generalization/specialization principles. This gives rise to ISA relationships between decision requirements. Finally, a complex DR is composed of other simpler decision



**Fig. 5.** An Abstract Decision Requirement ISA Hierarchy

requirements. Complex decision requirements form an AND/OR hierarchy in a manner analogous to AND/OR goal hierarchies of goal-oriented Requirements Engineering techniques.

To illustrate an abstract DR, consider an automobile plant that makes 1-tonne and 13-tonne trucks (Fig. 5). Let the decision of interest be *Set up New Assembly Line* and the required information be *Unsatisfied Orders*. The DR is shown in Fig. 5. 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*.



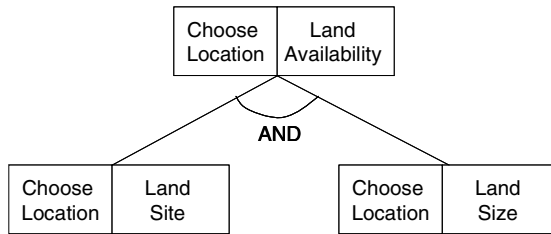
**Fig. 6.** Composition of Decision Requirements

Now let us illustrate a complex DR. The Decision Requirement  $\langle \textit{Set up New Assembly Line}, \textit{Unsatisfied Orders} \rangle$  is a complex one having two component decision requirements,  $\langle \textit{Decide Capacity}, \textit{Resources Available} \rangle$  and  $\langle \textit{Choose Location}, \textit{Land Availability} \rangle$ . An AND link connects these two components so as to define the complex decision requirement,  $\langle \textit{Set up New Assembly Line}, \textit{Unsatisfied Orders} \rangle$  (see Fig. 6).

Similar to AND composition, we can have OR composition. For example, when setting up a new assembly line, the question of whether or not it shall be part of an

existing profit centre has to be addressed. This gives us the decision requirements (Fig. 6), <Create Separate Profit Centre, Staff Availability> and <Merge with existing Profit Centre, Volume of Work Handled>. These two are in an OR relationship with one another. Notice that we have arrived at this through decision composition as well.

Aside from showing AND and OR decomposition of decisions, the foregoing also 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. 6 is shown as associated with the information, Land Availability. Land availability can be decomposed into two pieces of information, Land site and Land size

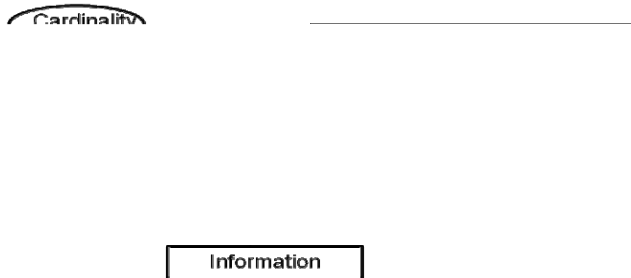


**Fig. 7.** Information Decomposition for DR Decomposition

Then the complex DR <Choose location, Land availability> can be decomposed into <Choose Location, Land site> and <Choose Location, Land size> respectively as shown in Fig.7.

### 3.2 Meta-model of Decisions

The key concept underlying the decision meta model of Fig. 8 is that of a decision parameter. Decision parameters denote different aspects of a decision. They reveal the factors that must be taken into consideration before a decision can be selected by the decision maker.



**Fig. 8.** Decision Meta Model



The decision to decision parameter relationship is M:N. There can be many decision parameters for a decision and a given decision parameter can be associated with more than one decision. A decision must have at least one decision parameter associated with it. This is because the absence of a decision parameter implies that there is no factor having a bearing to the decision. Therefore there is nothing to consider in taking the decision and it can be taken spontaneously. Such irrational decisions do not contribute to our objective of discovering the information to be maintained in the Data Warehouse and are therefore, out of our scope. Fig. 8 also shows that a decision parameter must be associated with at least one decision. This is to avoid irrelevant decision parameters from cluttering up the requirements engineering process.

It is possible to classify decision parameters in two different ways. According to one classification, a decision parameter may be dependent or independent. According to the second, a decision parameter may be a decision or information. We consider each of these in turn.

Dependent parameters depend on other, basic parameters for their existence. Those parameters that amplify the properties of other parameters are said to be dependent parameters. If the basic parameter becomes relevant or irrelevant, then parameters dependent on it also become relevant or irrelevant respectively. In contrast, independent parameters are orthogonal to one another. Each such parameter determines 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.

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.

Now, consider the second typology of a decision parameter. A decision parameter which is a decision represents the decomposition property of decisions that is used for DR decomposition. It shows the decomposition structure of the decision. On the other hand a decision parameter that is information behaves as a specification of the data to be kept in the data warehouse To-Be.

In this paper we have given a graphical notation for the decision requirements and their structure. In [17] we have introduced a dependency graph to represent decision parameters and dependencies between them. We used this graph there to distinguish between those parameters that are decision and information (details of information are given in the next section) respectively.

### 3.3 Modeling Information

We have developed in Fig 9, our model of information from the point of view of Data Warehouse support for decision making. We are interested in three kinds of information, that which exists in the most detailed form, summarized information or aggregates, and historical information. Aggregate information is obtained as a summary by computing from simpler information. This is shown in Fig. 9, 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, in the attribute Period.

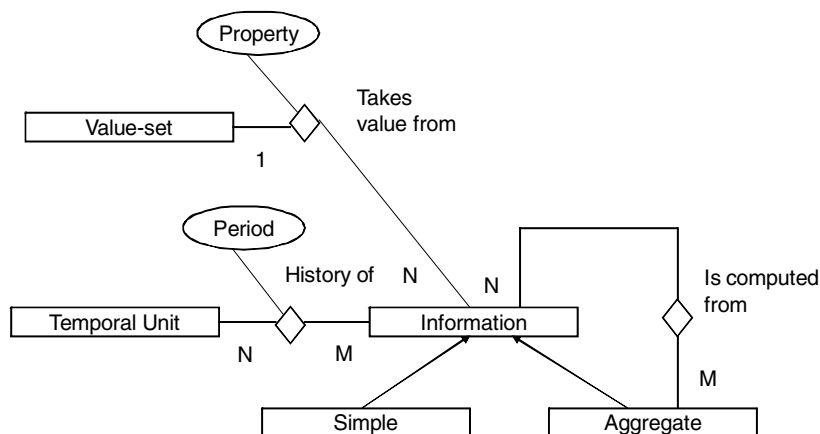


Fig. 9. Model of Information in Data Warehouses

Information is also associated with a value-set and takes on values from it. In Fig. 9 this association is called “Takes value from”.

#### 4 Case Study: A Health Scheme

The ideas presented here were applied to a health scheme operating in India. There are a number of dispensaries for providing medical services to members of the health scheme. There are well defined criteria for determining who can and who cannot be members. All eligible people may exercise the choice of obtaining membership of the health scheme and paying a fixed monthly charge. Once membership is granted, the beneficiary may go to any dispensary under the scheme and register for obtaining services there. The registrant may change dispensaries or discontinue membership in the scheme at will.

A dispensary is staffed with doctors who may refer registrants to other hospitals and specialists. A dispensary has in-house pharmacy with staff to disburse medicine. If a prescribed medicine is not available with this pharmacy, then a registrant may buy it from a notified list of external pharmacies and the costs shall be reimbursed. It is possible to obtain services of laboratories, radiology etc.

The information system provides a number of functions. For example, there are actions to consult a doctor, issue medicine, refer to a hospital, undergo tests etc. These functions are directly aimed at providing services to registrants. Additionally, there

are functions to induct a new doctor, add an external pharmacy to the notified list, empanel specialists and hospitals etc.

Imperative decisions related to health care can be made at the dispensary. For example, consider the tactical decision from the choice set {stock medicine in pharmacy of dispensary, obtain from central stores on need, allow medicine from external pharmacies}. A study of patterns of occurrence of health problems, fast moving/slow moving stock, and existing storage facilities leads to deciding what is to be stocked, what is to be obtained from the central store and what is to be allowed from external pharmacies. Once tactical decisions are made then operational decisions are made about orders for individual medicines and quantities. The information system is then invoked for indents, purchase and deliveries to be posted.

As another example, consider providing specialized advice. The dispensary makes an analysis of occurrence of health problems, ability to service them, patient convenience and other such factors. As a result, it may identify the need for some specialist medical advice not conveniently available in the dispensary. This constitutes an imperative decision of the operational kind. However, since the dispensary does not have the authority to do this, it makes a recommendation to headquarters about the need for the new specialization. Headquarters makes an administrative decision of the managerial kind, identifies the facility, and communicates it to the dispensary which, in turn, updates its information system.

Now let us consider managerial decisions. In the health scheme, these are essentially taken by headquarters. In doing so, requirements of dispensaries are taken into account as brought out in the example above.

A major managerial decision is about empanelment of medical facilities. Headquarters makes periodic calls for applications from hospitals for empanelment. The decision requirements are as shown in Fig. 10. There can be three kinds of hospitals, general (Gen.), super specialized (SS), and AYUSH. The last are those providing Indian systems of medicine (Ayurveda, Yoga, Naturopathy and Unani). For deciding which to empanel, headquarters needs information about available facilities. Thus, we get the DRs as shown in Fig. 10. There is an abstract DR <empanel hospital, facilities> which is specialized into the three DRs for the different kinds of hospitals.

Facilities comprise three aspects, space, equipment, and staff. Thus, <Empanel hospital, facilities> is an aggregate of the three DRs shown in Fig. 11, namely,

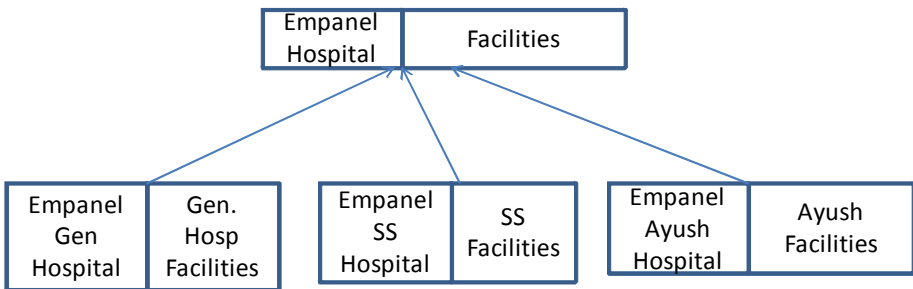
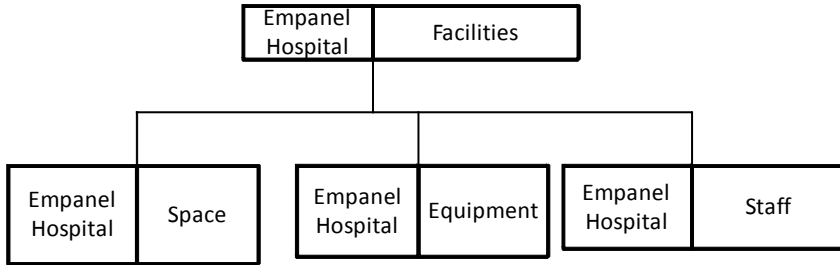


Fig. 10. The abstract DR < hospital, facilities>

<empanel hospital, space>, <empanel hospital, equipment> and <empanel hospital, staff>. Space information consists of sizes of wards, sizes of rooms, pharmacy stores and other Ayush spaces [14] and occupancy rates; equipment is the laboratory equipment, operating theatre equipment, number of beds and their occupancy ratio etc. and their use; staff information consists of doctors, paramedics, nurses and others [14] with their qualifications, the number and nature of cases/patients handled by them and other personal information [14].



**Fig. 11.** A complex DR

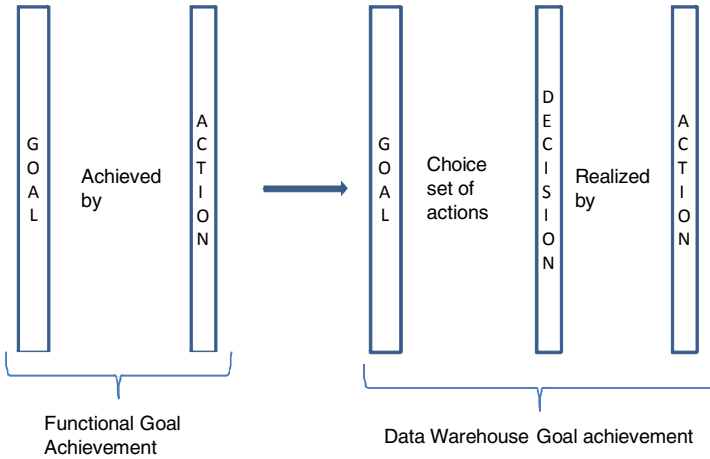
Now, headquarters defines norms and standards that must be met by all empanelled hospitals. These take the form of specified equipment, minimum requirements, and various norms. We give below examples of these for the Ayush class of hospitals. The full details of the case can be found in [14].

- Snehan Room 100 Sq. ft.
- Sirodhara Yantra & appliances 2 Nos.
- Vasti Yantra
- Droni/Bath tub
- O.P.D. 300 Sq. ft.
- Pharmacy/Store 300 Sq. ft
- Resident Medical Officer 1
- Masseurs/Panchakarma Attendants 4
- Staff Nurses 4 (Round the clock)
- Pharmacist 2
- Bed occupancy rate (Norm 50%)
- Bed:Nurse ratio for general bed 8:1

Whereas administrative decisions are taken after considering the extent to which given norms and standards are met, policy decisions define these. The elicitation of these norms is an unaddressed issue.

## 5 Related Works

First, let us consider the notion of goal in goal oriented requirements engineering approaches and compare it with our proposal. On the left hand side of Fig 12 we show



**Fig. 12.** Functional and Data Warehouse Goal orientation

the former. The notion here is of a goal being achieved by an action, a function. This helps in specifying system functionality. No support is provided to the user in determining which of the many actions is to be performed at a given moment. In our proposals, however, the focus is on determining the most suitable alternative, by examining the state of the organization as reflected in the data warehouse. Once this is done, then the action corresponding to the decision is invoked.

Now consider our approach with respect to database and ER driven approaches compared in Table 1. It can be seen that the decision requirement driven approach does not attempt to directly reach facts and dimensions like the others. Additionally, unlike the data base and ER approaches, the decision requirement approach can identify aggregate and historical information to be maintained. Finally, from a conceptual point of view, there is an explicit notion of a decision and decision requirements modeling which lays the basis for data warehouse development. Evidently, our approach lies upstream of the others.

Now, let us compare our proposals with goal oriented approaches in data warehouse development: Böhnlein and Ulbrich-vom Ende [6,7], Golfarelli and Rizzi [15] and Bonifati et al [16]. The first of these determines measures and dimensions from an initial study of goals and services to be provided by the organization. The second extends Tropos to requirements engineering of Data Warehouses. In the organizational perspective (a) an actor diagram is built (b) goals of actors are identified for example, ‘manage a/c transactions’ and (c) facts are associated to goals. Goals are decomposed using goal decomposition techniques. Thereafter, in the decisional perspective, the actor diagram is analyzed and each fact is related to its dimensions and a set of measures are associated with facts. The last proposal builds data marts by determining goals using the Goal-Question-Metric approach. The result is aggregated and refined and ideal star schemata are extracted. Then it uses ER schemata of operational data bases and extracts candidate star schemata. The ideal star schemata are matched with these and candidate star schemata are ranked according to the metrics for selection.

**Table 1.** A Comparison

	<b>Database driven</b>	<b>ER driven</b>	<b>Decision Requirement driven</b>
<b>Facts</b>	Yes, manually	Yes, manually	No
<b>Dimensions</b>	Yes, manually	Yes, algorithmically	No
<b>Star schemas</b>	Yes, manually	Yes, manually	No
<b>Aggregates</b>	Manually. No guidance	Manually. No guidance	Yes as Information property
<b>History</b>	Manually. No guidance	Manually. No guidance	Yes, as information property
<b>Decision Capacity</b>	Not articulated	Not articulated	Yes

Again, notice that [6,7] and [16] reach data warehouse contents directly from goals without an explicit decisional stage. In this respect the approach of [15] goes forward and recognizes the need to do further analysis from the decisional point of view. In our case, this movement has been taken to its logical conclusion. We explicitly model the full decision making capability and associated information requirements. In earlier work, we have shown how information associated with decisions can be elicited [17]. This information is obtained at a high level and is relatively unstructured. It is made more structured through the information scenario technique of [13] which produces contents of the data warehouse To-Be in the form of an ER diagram. Thereafter, the ER diagram is converted into star schemata using available algorithms. Thus, it can be seen that our proposal is to fully investigate the decision-information association before proceeding to development of star schemata.

GRAI grid [18] gives an overview of “decision making procedures and informational flows within a production control system”. It provides an architecture of decision making system. Decisional processes occur at the lowest level and are associated with time. GRAI models repetitive and ‘period-driven’ decisions. In our case, the decision making procedure per se is not interesting. Rather, it is the decision-information association that is of interest because it helps us to define data warehouse contents and thereby makes available information required for decision making. The procedure of decision making and its effect on the organization is outside our scope.

Finally, our decisional environment is similar to the work system proposed in [19]. However, it addresses decision making, not operational information systems

## 6 Conclusion

The notion of decision making is fundamental to data warehouses. It implies the existence of a choice set from which the best alternative, one that meets organizational

goals, is selected. We have shown that these alternatives can be (a) managerial, for setting up the environment for providing service and (b) imperative, for providing the right service. We have placed emphasis 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.

Future work is centered 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* 18(6), 483–497 (1992)
2. van Lamsweerde, A.: Goal-Oriented Requirements Engineering: A Guided Tour. Invited Paper for RE 2001 - 5th IEEE International Symposium on Requirements Engineering, Toronto, pp. 249–263 (August 2001)
3. Sutcliffe, A., Maiden, N.: Bridging the Requirements Gap: Policies, Goals and Domains. In: *Proc. IWSSD-7 - 7th Intl. Workshop on Software Specification and Design*. IEEE, Los Alamitos (1993)
4. Yu, E.S.K.: Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. In: *Proc. Third IEEE Symposium on Requirements Engineering*, pp. 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. In: *Proc. Of Workshop on Data Warehousing and OLAP (DOPLAP), USA*, pp. 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, Heidelberg (2000)
8. Rilson, F., Paim, S., Castro, J.F.B.: DWARF: An Approach for Requirements Definition and Management of Data Warehouse Systems. In: *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. In: *Proceedings of Workshop on REFSQ*, pp. 75–82 (2003)
10. Prakash, N., Gosain, A.: Requirements Driven Data Warehouse Development. In: Eder, J., Missikoff, M. (eds.) *CAiSE 2003*. LNCS, vol. 2681, pp. 13–17. Springer, Heidelberg (2003)
11. Prakash, N., Singh, Y., Gosain, A.: Informational Scenarios for Data Warehouse Requirements Elicitation. In: Atzeni, P., Chu, W., Lu, H., Zhou, S., Ling, T.-W. (eds.) *ER 2004*. LNCS, vol. 3288, pp. 205–216. Springer, Heidelberg (2004)
12. Winter, R., Strauch, B.: Information Requirements Engineering for Data Warehouse Systems. In: *ACM Symposium on Applied Computing, Cyprus*, pp. 1359–1365 (2004)
13. Prakash, N., Gosain, A.: An Approach to Engineering the Requirements of Data Warehouses. *Requirements Engineering Journal* 13(1), 49–72 (2008)
14. Government of India, Department of Ayush, Ministry of Health and Family Welfare, No.Z.20018/4/2000-ISM (Tech)/HD Cell, National Competitive Bidding

15. Golfarelli, M., Rizzi, S.: Designing the Data Warehouse: Key Steps and Crucial Issues. *Journal of Computer Science and Information Management* 2(3) (1999)
16. Bonifati, A., Cattaneo, F., Ceri, S., Fuggetta, A., Paraboschi, S.: Designing Data Marts for Data Warehouses. *ACM Trans. Softw. Eng. Methodol.* 10(4), 452–483 (2001)
17. Prakash, N., Prakash, D., Sharma, Y.K.: Towards Better Fitting Data Warehouse Systems. In: Persson, A., Stirna, J. (eds.) *PoEM 2009. LNBIP*, vol. 39, pp. 130–144 (2009)
18. Carrie, A.S., Macintosh, R.: An Assessment of GRAI Grids and their use in the Strathclyde Integration Method, *production Planning and Control*. vol. 8(2), pp: 106–113 (1997)
19. Alter, S.: A General yet Useful Theory of Information Systems. *Comm. of Association for Information Systems* 1(13), 1–68 (1999)
20. Inmon, W.H.: *Building the Data Warehouse*. John Wiley and Sons, Chichester (1996)