INDRAPRASTHA INSTITUTE of INFORMATION TECHNOLOGY DELHI

Capstone Project on Topic

Decision Backlog Selection for Agile Data Warehouse Development

Submitted by

Name: SWATI VERMA

Roll No.: MT19073

Under the Supervision of Dr. NAVEEN PRAKASH

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING IIIT-DELHI

December 2020

Acknowledgments

For completing my project, I took the help and guidance of some of the respected persons who deserve my greatest gratitude.

I would like to express my special thanks of gratitude to Dr. Naveen Prakash , my project guide who gave me the opportunity to do this project under his wonderful guidance. This project really helped me a lot in understanding the concepts of Data Warehousing.

I would also like to expand my deepest gratitude to all the members of Sigma Enterprise for helping me out throughout this project. Without their involvement this project would not have been successful.

My special thanks and appreciation goes to Pragya Dara for being a wonderful partner who willingly helped me with all her abilities throughout the project.

At last, I would like to thank my parents and friends who have directly or indirectly guided me during this whole project.

TABLE OF CONTENTS

1. **What is a Data Warehouse?**
   1.1 The Inmon Approach
      1.1.1 Subject Oriented
      1.1.2 Time-Variant
      1.1.3 Non-volatile
      1.1.4 Integrated
   1.2 The Kimball Approach
      1.2.1 The Kimball DW/BI Lifecycle
         1.2.1.1 Iterative Model
         1.2.1.2 Incremental Model
         1.2.1.3 Iterative- Incremental Approach used by Kimball
      1.2.2 Drawbacks of Kimball's Approach
   1.3 The Decision-Making Perspective
      1.3.1 Herbert Simon Model on Decision-Making
         1.3.1.1 Intelligence Phase
         1.3.1.2 Design Phase
         1.3.1.3 Choice Phase
2. **Sigma Enterprise Description**
   2.1 Introduction to Sigma Enterprise
   2.2 The workflow of Sigma Enterprise
      2.2.1 Organizational chart
   2.3 Roles of Family Members
   2.4 Decisions of Sigma Enterprise
      2.4.1 Description of Decisions
   2.5 Composite Decisions of Sigma Enterprise
   2.6 Decision Dependency Graph of Sigma Enterprise
3. **Decision Information Elicitation Model**
   3.1 Influence Diagram
      3.1.1 Decision Node
      3.1.2 Uncertainty Nodes
      3.1.3 Value Nodes
      3.1.4 Functional Edges
      3.1.5 Informational Edges
   3.2 DIEM Components
      3.2.1 Decisions
      3.2.2 Actions

3.2.3 Uncertainties
3.2.4 Objectives
3.3 DIEM components for Sigma Enterprise
3.4 DIEM tool for Sigma Enterprise
4. Decision Backlog
4.1 Backlog of Decisions
4.1.1 Free Backlog
4.1.2 Specialized Backlog
4.1.3 Composite Backlog
4.1.4 Mixed Backlog
4.2 Selecting Decisions from Free Backlog
4.3 Selecting Decisions from Specialized Backlog
4.4 Selecting Decisions from Composite Backlog
4.5 Selecting Decisions from Mixed Backlog
4.6 Selecting Decisions from Decision Backlog Tool for Sigma Enterprise
5. Information Backlog
5.1 The Intelligence Phase
5.2 The Choice Phase
5.3 Information Backlog Tool for Sigma Enterprise
6. Description of Tables
7. Conclusion
7.1 Comparison with Kimball's approach of Data warehouse design
7.2 Applying Agile Backlog Approach to Sigma
7.3 Challenges Faced
7.4 Future Work
List of References
Appendix How to install and Run the Tool

LIST OF FIGURES

Figure No.	Figure Name	Page No.
1.1	Inmon’s Model for Data Warehouse	1
1.2	Kimball’s Model for Data Warehouse	3
1.3	Simon’s Model on Decision Making	5
2.1	Organizational chart of Sigma Enterprise	9
2.2	Composite Decision 1	15
2.3	Composite Decision 2	15
2.4	Composite Decision 3	16
2.5	Dependency Graph for Sigma Enterprise	16
3.1-3.19	Influence Diagrams	17-34
3.20-3.30	UI Screenshots for DIEM Tool	35-40
4.1	Free Backlog Decision Graph	42
4.2	Specialized Decisions Hierarchy	43
4.3	Composite Decisions Hierarchy	44
4.4-4.15	UI Screenshots for Decision Backlog Tool	45-50
5.1-5.9	UI Screenshots for Information Backlog Tool	52-56

1. What is a Data Warehouse?

1.1 The Inmon Approach:

Bill Inmon introduced the top-down approach for building data warehouses which starts with building a centralized warehouse for an enterprise where all the available data from heterogeneous resources are merged into a subject-oriented, non-volatile, integrated, and time-variant collection of data that can be used to make strategic decisions by an enterprise.[4] Data marts are created only after the data warehouse has been created and are used by the departments within the enterprise for analytical purposes.

The transactional database changes frequently on a daily basis whenever a transaction takes place and hence has no data for analysis or decision making. But a data warehouse provides Online Analytical Processing (OLAP) tools that help in decision making and also gives summarized, generalized and consolidated data in a multidimensional view.

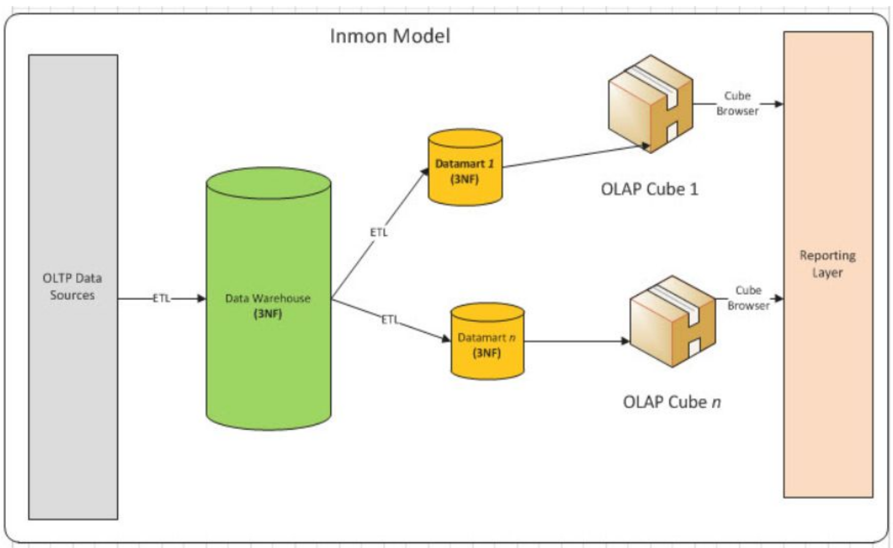

Fig 1.1 Inmon's Model for Data Warehouse

The key features of data warehousing described by Inmon are as follows:

1.1.1 Subject Oriented: Rather than focusing on the operations going on in an organization, data warehousing focuses on a subject and provides all the relevant information related to that subject. For example, if an organization wants to decide from " which supplier they want to buy products", so in this case, the subject would be "supplier".

1.1.2 Time-Variant: All the data in the data warehouse is associated with time stamps. Data warehouse stores historical data and as soon as some changes are made, the data warehouse does not update it immediately. The Data warehouse has a refreshing time at which all changes are updated.

1.1.3 Non-volatile: Data in data warehouses are not changed frequently. Old data is not erased and kept as it is when the new data is added. Data warehouses are highly read oriented and no data is lost. Periodic refreshment is done in which irrelevant data are omitted, error correction and update operations are performed.

1.1.4 Integrated: Data warehouse contains data from various heterogeneous resources such as flat files, relational databases, etc. Unit differences such as height in feet, centimeters, meters, data representation differences such as sex of a person as male/female, m/f, 0/1, etc. are standardized to enhance the effective analysis of data. As different departments of the same organization store data as per their convenience, it is the responsibility of the data warehouse to integrate them

Although Inmon's approach has various advantages like data in the warehouse are normalized and easy to maintain, there are many places in which this approach does not perform well.[5] Disadvantages of Inmon's model are as follows:

1. It cannot be used in cases that require quick setup and delivery time because here we have to build a whole data warehouse first which can be very time-consuming.
2. This approach requires a large specialized team to build a warehouse, so if the team size is less then this approach will not work efficiently
3. Undoubtedly, the bigger the business organization, the more data it contains. Building a data warehouse for such an organization is a complex process and requires a lot of time and initial set-up costs.
4. It focuses on enterprise-wide data rather than small department areas.

So, to overcome the disadvantages of this approach, Kimball introduced a bottom-up approach for building data warehouses.

1.2 The Kimball Approach

Ralph Kimball defines the data warehouse as "a copy of transaction data specifically structured for query and analysis".[6] He proposed the bottom-up approach for building data warehouses which starts by identifying the key needs of the business processes and questions that the data warehouse needs to answer. The data marts are formed first based on the requirements of the departments and then integrated to form a comprehensive data warehouse. Data marts are basically a subset of a data warehouse that keeps the information about a single department rather than keeping information about the whole enterprise.

Kimball's model presents an incremental approach to build an enterprise DW/BI system called the enterprise data warehouse bus architecture. This architecture is composed of a set of tightly integrated data marts that pulls out their power from a common set of conformed facts and dimensions. The facts contain a quantitative or measurable piece of information about the business process while dimensions are the companions to the fact which describes the entities of the fact table. The dimensions in bus architecture are shared between facts in two or more data marts which increases the reusability of dimensions.

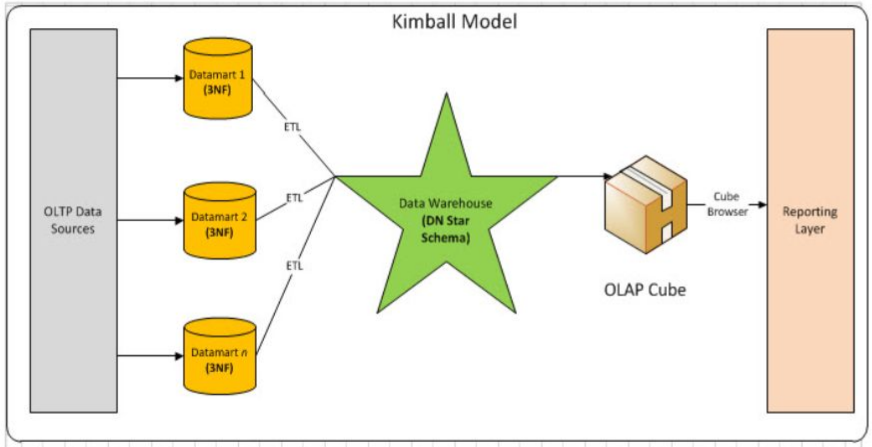

Fig 1.2 Kimball's Model for Data Warehouse

1.2.1 The Kimball DW/BI Lifecycle:

The Kimball model starts with identifying the key business requirements and focusing on how to add the best value to the enterprise. The enterprise performs an initial set of interviews to gather and prioritize business needs. Once the list of business processes and their associated information has been identified and prioritized, the next step is to take the highest priority process and collect all the detailed information associated with it.[7]

Once this detailed information is gathered, the next phase is to start implementing it. This phase includes various steps. Initially, the functionalities and associated tools and technologies that would help in meeting the business requirements are identified. In the next step, a logical model that can be used to support analytical requirements is decided. Kimball uses a dimensional model. Once the model is selected, the team starts building the target database. In the next step, the Extract, Transform, and Load (ETL) system is created which is used to populate data into databases. The last step deals with the BI application which provides the initial set of analysis and detailed reports which will deliver the first business value to the enterprise.

After the implementation part is done, the deployment phase starts which deploys reports, query tools, and final application to the user community. Extensive communication, documentation, training, and support are needed in this phase.

Once the deployment is done, the changes which are taken as feedback from the previous iteration are incorporated and the analytics and designing teams again start gathering detailed information about the next highest priority business need and start the same process all over again.

The Kimball approach leads to the development of a combined iterative-incremental approach for building a data warehouse. A brief description of an iterative, incremental, and a combined approach are discussed below:

1.2.1.1 Iterative Model: The iterative model is a software development process that focuses on initial, simplified implementation and iteratively upgrades the evolving version until the full system is completed and ready to be deployed. Every release of this model finishes in an exact and fixed period called iteration. The iterative model provides feedback paths from every phase to its preceding phase in which changes can be made accordingly. The iterative model is used when the requirements are clearly and completely understood or when functionalities evolve over time. But this approach is not useful when the project size is too small because ever-changing needs may cause over budget. Customers have to wait until the full product is developed.[2]

1.2.1.2 Incremental Model: An incremental model is a software development process where the requirements are broken into multiple independent modules, which are incrementally constructed and delivered. Each module goes through all the phases of the software development life cycle, which includes requirement gathering, analysis and design, implementation, testing/validating, and maintenance phase. Initially, the core features of the software are developed such that it can be in a deliverable state. Once the main features are developed, additional features are added to the software's subsequent release until the software has been fully implemented. After each successive version is implemented and delivered, feedback is taken from the customer, and the new features are incorporated in the next version.[3] So, we can say that each version of the product has more features over the previous ones. This model overcomes some of the disadvantages of the iterative approach like in this approach, the customer does not have to wait till the final product is developed. The higher priority requirements are implemented first so the initial release time is very less. But this method has its own disadvantages like, if the increment itself is extensive then the initial delivery time will again be high.

1.2.1.3 Iterative-Incremental Approach used by Kimball: This approach takes the good points from both the models. The Kimball approach promotes the development of a combined iterative-incremental model that consists of multiple increments with each increment having various iterations. In the incremental model, all the increment requirements are gathered first, and then the module is developed. If the increment requirements are extensive, then it would take more time to release the increment. So, to cut down the delivery time and initial cost of an increment, it is best to release the increment in the small iterations. This approach does not even require a big team of specialists for building data marts or increments and is very cost-effective.

1.2.2 Drawback of Kimball's Approach:

Kimball's approach has two major disadvantages:

1. There is no information about how the data marts are prioritized and which data mart should be developed first
2. The lead delivery time is still very high.

1.3 The Decision-Making Perspective

A decision can be defined as a commitment to action, course of action, or a strategy. As the data warehouse contains historical data, it can help in the decision-making process within an organization. It helps enterprises to take strategic decisions on the basis of information associated with past decisions.

P. Uhrowczik in terms of the decision-making process defines data warehousing as " The data, process, tools, and facilities used to manage and deliver complete, timely, accurate and understandable business information to authorized individuals for effective decision making". The decision-maker should have complete knowledge about all the decisions and business information of an organization. The information should be timely i.e. at the time of decision making all the information should be present. There should not be any error during information collection. All the information required during decision making should be accurate and easy to understand.

1.3.1 Herbert Simon Model on Decision-Making

Herbert Simon, one of the key researchers in the field of decision making described a decision-making process as three steps process:

1.3.1.1 Intelligence Phase: In this phase, initially, the environment or the current situation of an organization is analyzed by the decision-maker, and problems that require decision-making are identified. This phase also deals with the data collection which helps the decision-maker in a clear and complete understanding of the problem.[9]

1.3.1.2 Design Phase: In this phase, different alternatives for the problems identified in the intelligent phase are outlined, designed, and evaluated. Evaluation of different alternatives are performed on the basis of criteria decided in the intelligent phase.

1.3.1.3 Choice Phase: This phase deals with selecting the best alternation among the different alternatives available.[10]

All the above steps are implemented at the end which decides the success or failure of the process. If the result is a success then the required solution is obtained else again the process is started all over again by going into the previous phases.[11]

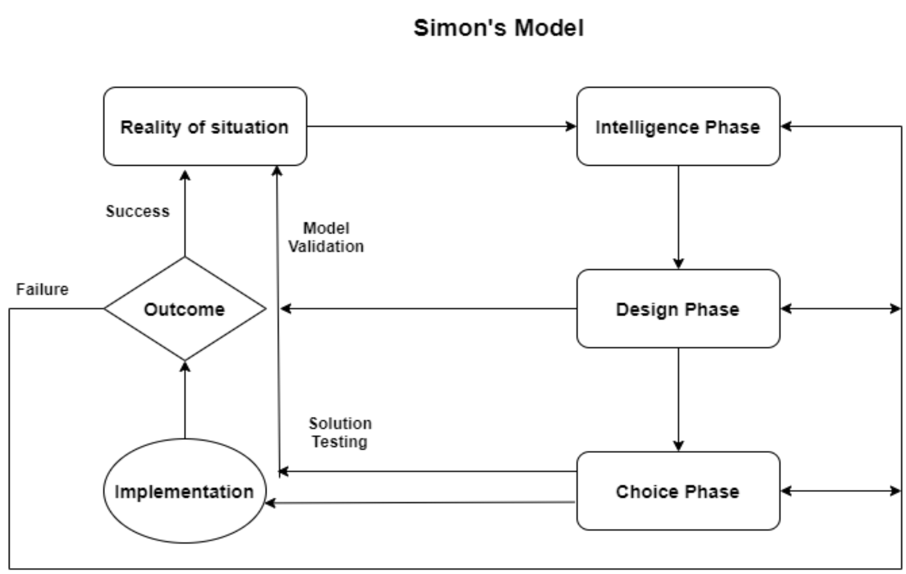

Fig 1.3 Simon's Model on Decision Making

Let us take an example to understand Simon's model:

Suppose a person wants to travel from Delhi to Bangalore via airplane. Then, in the intelligence phase, the person would collect all the information needed to travel like which airlines are present that day, schedules of the airlines, etc. Evaluation criteria like the number of halts taken by each airline, total travel time, cost of the airline, comfort, and safety provided by the airlines are also decided in this phase.

In the design phase, different alternatives are evaluated on the basis of some criteria defined in the intelligent phase.

Criteria	Airline 1	Airline 2	Airline3
Cost	Rs. 6700	Rs. 3500	Rs. 4700
Time	2 hours	6 hours	3 hours
Safety and comfort rating	8.5	6.5	7.0

In the choice phase, the person will select the best airline which meets the user's criteria. Like if a person wants to minimize travel costs and does not care about the traveling time and comfort the user would go for Airline 2. If a person wants less traveling time then the best choice would be Airline 1.

The decision problem is built during the design phase of Simon's model.[12] There are mainly two ways of representing a decision problem. The first one is a decision tree where each node consists of some test conditions and the edges represent the possible outcome or action and can be viewed as a flow chart. The second one is an influence diagram which represents the more global view of the decision problem. It provides a graphical way to understand the factors involved in a decision-making process. This global picture provides a better understanding of decision problems and will help to select decisions from a backlog of decisions. In this project, the components of the influence diagram are captured for the Sigma Enterprise which will be discussed in the later chapters.

As we know, the bigger the organization, the more information it contains regarding various departments. Building data warehouses for such big organizations can be very complex and costly. It also takes too much time to build such data warehouses which lead to longer delivery time. To reduce the lead time, Kimbell promoted the incremental-iterative approach for building a data warehouse but this approach also has several drawbacks which are explained in the earlier chapter. To overcome the drawbacks of Kimball's approach this paper proposes a solution to the problem of selecting which data marts are to be developed first.

The paper presents a decisional approach to build a data warehouse. From a decision-oriented perspective, an enterprise can be seen as having a large number of decisions. Building a data warehouse for such an enterprise would be very complex and time-consuming. So, to reduce the lead time, a combined iterative-incremental approach has been used in this project where each decision is considered as an increment, and the actions, uncertainties, and objectives corresponding to that decision are taken as iterations within that increment. The decisions are taken incrementally, one decision at a time. Our approach first starts with selecting a decision from a backlog of decisions and then selecting the information from the backlog of information relevant to that decision. In this paper, an assumption is taken that a decision is selected on the basis of a syntactic or semantic priority. Syntactic priority informs the number of decisions that must be supported for a given decision to be supported and semantic priority is decided by the stakeholders of the data warehouse that which decision should be given higher priority and which decision should be given lower priority. Once the highest priority decision and its relevant information are selected then the next higher priority decision is considered and so on till all the decisions are explored.

At the end of this project, the tool developed for selecting decisions is checked for the decisions of Sigma Enterprise which is a Faridabad based family business. The next chapter contains a detailed description of the Sigma Enterprise and its decisions. Chapter 3 describes the influence diagram, the DIEM components captured for Sigma Enterprise, and the implementation of the DIEM tool. Chapter 4 contains information about different types of backlogs, the tool implemented for selecting a decision from a backlog of decisions, and testing the tool against the decisions of Sigma Enterprise. Chapter 5 describes the information backlog, the tool created for the intelligence and choice phase, and testing the tool against decisions of Sigma enterprise.

2. Sigma Enterprise Description

2.1 Introduction to Sigma Enterprise:

Sigma Enterprise is a family-based company located in Faridabad. It is a micro-enterprise that manufactures and delivers decorative components for fans like rings, trims, and canopies. The company is overlooked by three family members and has a staff strength of twenty employees, four supervisors, and three managers and are serving the fan industry for more than 30 years.

Sigma Enterprise has two divisions. The first division works as a plastic molding unit where plastic grains are heated and fed into cavities to get the required shape. The automated molding machine produces the required components and these components after cooling down are brought into the second division where they are coated and baked with different chemicals. Then vapor Aluminum is deposited to give these components different effects like shining or dull and different colors like golden, silver, bronze, etc. After the components are ready, they are packed and delivered to the customers.

2.2 The workflow of Sigma Enterprise:

Initially, the customers give their purchase order to the manager of the Sigma Enterprise. The purchase order consists of all the requirements of the customer such as what item they want, the number of items they want, etc. Whether to accept the order or not is decided by one of the family members. If the order cannot be accepted immediately then the family member decides whether to make the customer a potential customer or not else if the company does not manufacture the required component then the order is not accepted.

Once the order is accepted, the manager checks for the availability of the raw materials. The materials are restocked periodically for all future requirements. The raw materials include fine plastic grains, paints, polishes, and various chemicals.

If there are more than one order then the orders are prioritized and the production is planned accordingly.

Once the materials are issued and the production plan is finalized then the production begins. At present, there are eight manufacturing machines available in the factory. Production involves plastic molding, base coating, metalizing, and top coating.

After production is finished, quality checking is done to ensure that good quality products are delivered to the customer. After quality inspection, the products are packed and shipped to the customers. The payment follow-ups are done after the delivery of the items.

2.2.1 Organizational Chart

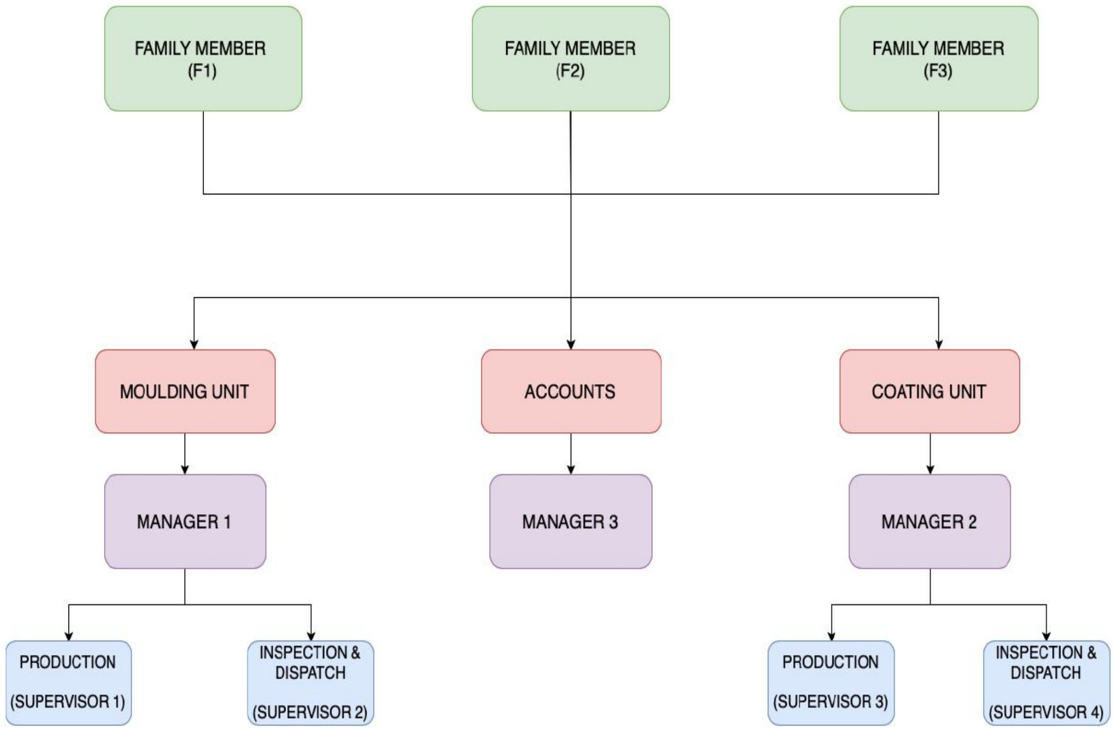

Fig 2.1 Organizational chart of Sigma Enterprise

2.3 Roles of Family Members:

The business is overlooked by three family members.

1. **Family Member 1**

- **Dealing with Clients:** The first family member deals with clients and is responsible for making decisions like whether to accept the order or not based on the requirements of the order. The member stays in touch with potential customers as well as current customers. If the order cannot be accepted immediately, then the customer becomes potential customers.
- **Production Planning:** Restocking of materials, dealing with suppliers from whom the materials are purchased, prioritizing the orders accordingly, and making production plans are done.
- **Dispatching Items:** The member is responsible for making marketing calls and dispatching the finished items to the customers.

2. **Family member 2**

- **Human Resources:** The second family member is responsible for hiring new employees, approving leaves, and other things related to employees.

- ❑ **Quality Management:** Responsible for checking the quality of manufactured items and also decides what to do with poor quality products.
- ❑ **Accounts:** Looks after the accounts of the company.

3. **Family member 3**

- ❑ **General Management:** The third family member is responsible for providing a better working environment and resource management.
- ❑ **Cost Reduction:** The member analyses and introduces new changes that would help the company in cost optimization with increasing efficiency, optimizing resource requirements, etc.

Responsibilities of Managers and Supervisors:

1. **Manager 1 (Molding unit/ Division 1):**
    - ❑ **Material Requirement:** On the basis of ongoing production and new orders, the manager is responsible for estimating raw material requirements for the first division i.e. for molding division, and also looks after the materials which are required to be restocked.
    - ❑ **Production Planning:** Manager 1 is responsible for making daily production plan machine wise as well as prioritizing the orders.
    - ❑ **Maintenance of the unit:** Manager 1 does periodic checking of the machines and is responsible for maintaining them so that the production of the items are not affected.
    - ❑ **Quality Checking:** The quality of items manufactured in division 1 is checked by the manager.

- **Supervisor 1:**
    - ❑ **Production:** The first supervisor supervises the daily production of all the machines of the molding unit and checks the stocks when the material requirements have been estimated.
- **Supervisor 2:**
    - ❑ **Inspection and Dispatching Items:** After the plastics are molded, quality inspection is supervised by the second supervisor and if items are of good quality then they are dispatched to the second division which belongs to the coating unit.

2. **Manager 2 (Coating Unit/ Division 2):**
    - ❑ **Material Requirement:** On the basis of ongoing production and new orders, the manager is responsible for estimating raw material requirements for the second division i.e. for coating division, and also looks after the materials which are required to be restocked.
    - ❑ **Production Planning:** The products received from the first division are coated and painted and which items to be coated first as well as machine wise production planning is done by the manager.

- **Maintenance of the unit:** Manager 2 does periodic checking of the machines and is responsible for maintaining them so that the production of the items are not affected.
- **Quality Checking:** The quality of finished items in division 2 is checked by the manager.

- **Supervisor 3:**
  - **Production:** Supervises daily production of the items in the coating unit and also estimates the paints and other chemical requirements for the coating unit.
- **Supervisor 4:**
  - **Inspection and Dispatching Items:** After the items are coated and painted, they are inspected for good quality by supervisor 4 and if the items are of good quality then they are packed and made ready for shipping.

3. **Manager 3 (Accounts) :**
   - Keeps records of all the invoices, payment history of customers, etc.
   - Manager 3 is responsible for collecting payments from various clients from time to time.
   - It is the responsibility of manager 3 to make payments to the suppliers of raw materials from time to time.

2.4 Decisions of Sigma Enterprise

Some of the decisions taken by the employees of Sigma Enterprise are shown in the below table:

F1-Family Member 1, F2-Family Member 2, F3- Family Member 3

M1-Manager 1(Molding unit), M2- Manager 2(Coating unit), M3-Manager 3(Accounts)

S1=Supervisor1(Molding unit), S2- Supervisor 2(Molding unit), S3-Supervisor 3(Coating unit), S4-Supervisor 4(Coating unit).

Decision#	Decision	Choice Set	Decision Maker	Stakeholders
1.	Whether to accept an order or not	1. Accept Order 2. Reject Order	F1	
2.	Can customer be a potential client	1. Add the customer to potential client/new clients list 2. Do not add the customer to the potential client list	F1	

3.	Modify production plan or not	1. Modify the existing plan. 2. Continue with an existing plan and later make a new one	F1	
4.	Which order to prioritize	Set priorities for all ongoing and new orders.	F1	M1, M2
5.	Whether restock is required or not	1. Supplies need to be restocked 2. Supplies do not need to be restocked	M1, M2	F1
6.	When to restock the supplies	Date of restocking	F1	M1, M2
7.	Which items to restock	List of various products that need to be restocked	M1, M2	F1
8.	Which suppliers to buy supplies from	List of suppliers from whom products can be purchased	F1	
9.	How much quantity to be restocked	No. of units of product required	M1, M2	F1
10.	How many units to be produced daily	No. of units to be produced on a daily basis	F1	M1, M2
11.	How many units to be produced per machine	No. of units to be produced on a daily basis by each machine	M1	
12.	Are more workers required	1. The current number of workers is sufficient. 2. More workers are required	F2	M1, M2
13.	What to do with poor quality products	1.Remold the plastic 2.Discard it	F2	M1, M2
14.	When to transport the products	Date of transporting the products	F1	S4
15.	Which mode of transportation should be used to transport the products	1.Truck 2.Car	F3	S4
16.	When to make marketing calls		F1	
17.	When to service a machine	Weekly/Monthly servicing schedule of machines	M1, M2	F1,F3

18.	Approve leave of an employee or not	1.Approve leave 2.Cancel leave request	F2	
19.	How many units to pack per box	No. of units of products to be packed per box	F3	S4

Table 1 Decisions of Sigma Enterprise

2.4.1 Description of Decisions:

1. **Whether to accept the order or not**
   Family member 1 takes the decision of whether to accept or reject the order received from various clients.
2. **Is the customer a potential client?**
   If the company manufactures the components ordered by the customer but current due to overload the order cannot be accepted then the family member 1 decides whether to make the customer a potential customer or not. But if the company does not manufacture the parts asked by the customer then the family member 1 cancels the order.
3. **Modify production plan or not**
   On the basis of new orders received, family member 1 decides whether to stick with the existing production plan or to come up with now plan.
4. **Which order to prioritize**
   Priorities of all the orders for production are set by family member 1.
5. **Restock is required or not**
   Managers of both the division decide whether supplies are needed to be restocked or not.
6. **When to restock the supplies**
   The date of restocking is decided by family member 1.
7. **Which products need to be restocked**
   Managers make the lists of materials which are needed to be restoked and give it to the family member 1.
8. **Which suppliers to buy supplies from**
   Among various suppliers, family member 1 decides from which supplier the materials can be purchased.
9. **How much quantity to be restocked**
   Managers decide the unit of products needed to be restocked in each division.
10. **How many units to be produced daily**
    Family member 1 decides how many units to be produced on a daily basis and asks managers to look after the daily production.
11. **How many units to be produced per machine**
    The manager of the molding unit takes the decision about the number of units produced on a daily basis machine-wise.
12. **Are more workers required**
    On the basis of orders received, family member 2 decides whether current number of

workers is sufficient or more workers are needed to be hired.

13. **What to do with poor quality products**
    Family member 2 does quality inspection and decides whether to remold the poor quality plastic products or to discard it.
14. **When to transport the products**
    The date of the shipment of the finished product is decided by family member 1.
15. **Which mode of transportation should be used to transport the products**
    Family member 3 decides whether to transport the items via car or truck.
16. **When to make marketing calls**
    Decisions related to marketing calls and advertisements are taken by family member 1.
17. **When to service a machine**
    Managers of division 1 and division 2 decide the schedule of servicing of machines like weekly, monthly, etc.
18. **Approve leave of an employee or not**
    Family member 2 decides whether to approve leave of an employee or cancel the leave.
19. **How many units to pack per box**
    Based on the size of boxes for packing, family member 3 decides how many units of items to be packed in each box.

2.5 Composite Decisions of Sigma Enterprise:

When multiple small decisions are taken in order to take any major decision, then such a decision that consists of a number of smaller decisions is known as a composite decision. In our project of Sigma Enterprise, we have considered multiple composite decisions.

The following are the composite decisions of Sigma Enterprise:

1) Modify the production plan or not?
    - ❏ Which order to prioritize?
    - ❏ How many units to be produced daily?
    - ❏ Extra employees are required or not?

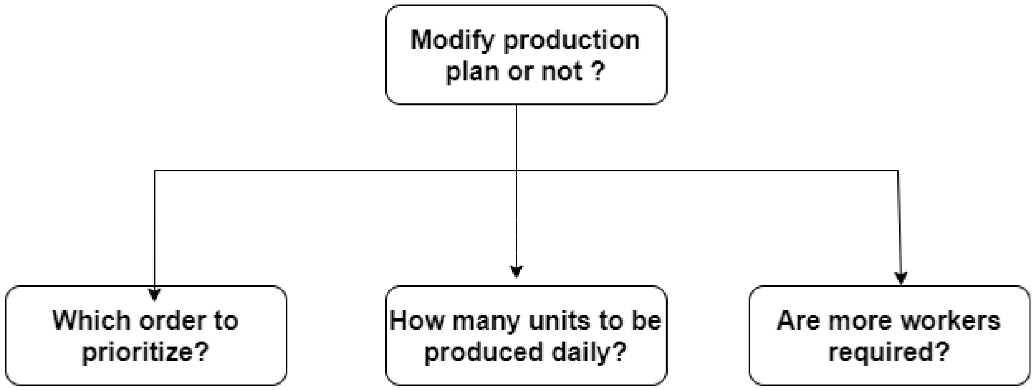

Fig 2.2 Composite Decision 1

2) Whether restock is required or not?
   - ❑ When to restock the supplies?
   - ❑ Which products need to be restocked?
   - ❑ Which suppliers to buy supplies from?
   - ❑ How much quantity to be restocked?

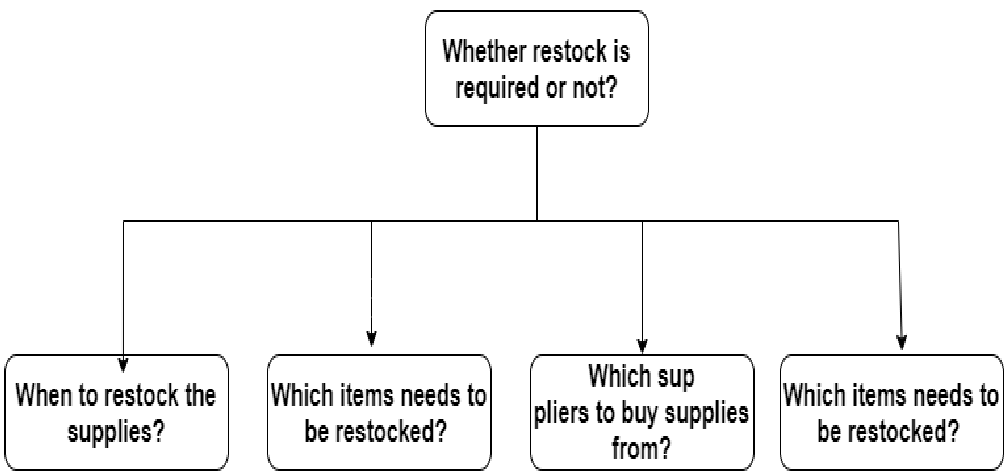

Fig 2.3 Composite Decision 2

3) When to transport products?
   - ❑ Which mode of transportation to use?
   - ❑ How many products to be packed per box?

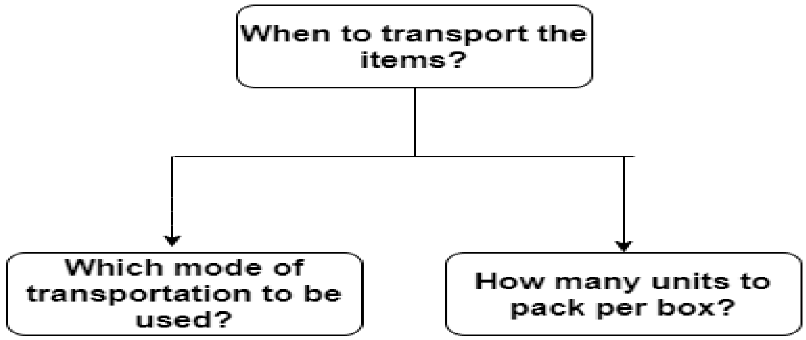

Fig 2.4 Composite Decision 3

2.6 Decision Dependency Graph of Sigma Enterprise:

Decision dependency graphs contain decisions in nodes and the edges represent the dependencies between them. In our project, we have considered logical dependencies. If A and B are two decisions and if A--->B then we can say decision B is logically dependent on decision A if the output of decision A affects decision B. If there is no edge between any two nodes then we can say both the decisions are independent of each other. This dependency graph is capture in the decision backlog tool described in the later chapter. It will help us to decide the sequence in which decisions should be selected from the backlog of decisions. The dependency decision graph of Sigma Enterprise is shown below:

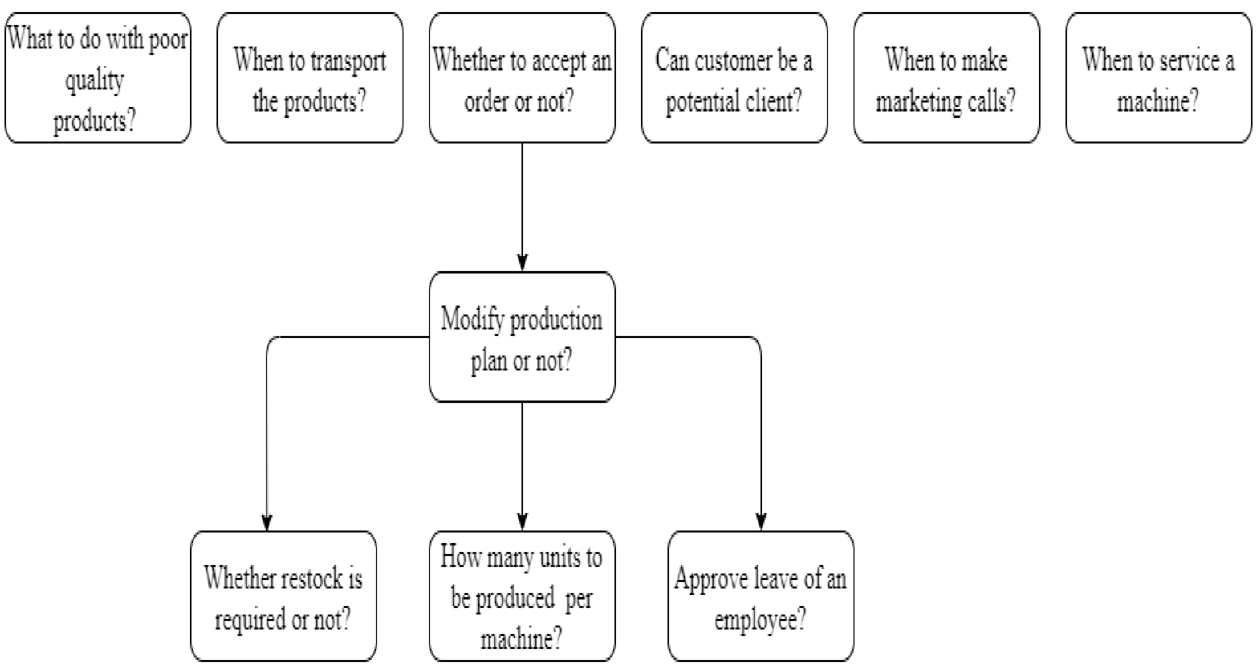

Fig 2.5 Dependency Graph for Sigma Enterprise

3. Decision Information Elicitation Model

3.1 Influence Diagram

An influence diagram is a directed acyclic graph which consists of three types of nodes and two types of edges. It represents the important elements such as decisions, outcomes, uncertainties and objectives as nodes with different shapes.

The nodes are described as follows:

3.1.1 Decision Nodes: This node represents the decision to be made or the choices that are available to the decision maker. Such nodes are represented using rectangular boxes.

3.1.2 Uncertainty Nodes: This node represents the uncertain quantities that a person cannot control. The decision maker does not know the value of these quantities because the complete information is not available at any particular time instant. They might get the complete information in the future. Such nodes are represented by ovals. One special type of uncertainty node is the deterministic node represented by a double oval whose outcome can be determined if the values of some other uncertainties are known.

3.1.3 Value Nodes: These are diamond shaped or rounded rectangular nodes which represent the outcome obtained by a decision.

The edges are:

3.1.4 Functional Edges: These are the solid arrows which either ends are value nodes or decision nodes. It shows that a node at the head of the arrow depends on information of the node present at the source of the arrow.

3.1.5 Informational Edges: These are dashed arrows which tell that the decision is made with the knowledge of the values or outcome of the source node.

3.2 DIEM Components

The DIEM model consists of concepts involved in the decision-making process. The concepts are:

3.2.1 Decisions: The decisions to be taken or the choices available.
3.2.2 Actions: For every decision an action is taken. For example, the decision whether to accept an order or not contains an action accept_reject_order.
3.2.3 Uncertainties: The quantities which are beyond control or which are not known.
3.2.4 Objectives: The outcomes of the decision.

3.3 DIEM components for Sigma Enterprise:

Decision 1: Whether to accept an order or not
Action: Accept/Reject Order

Uncertainties:

- Order Requirements
- Client
- Current Workload
- Expected Profit

Objectives:

- Maximize Profit
- Maximize Customer Satisfaction

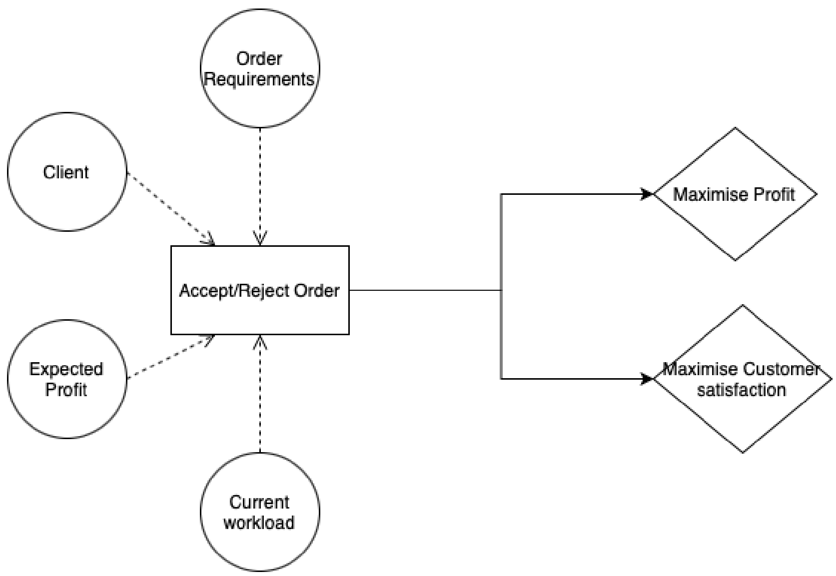

Fig 3.1 Influence Diagram for D1

Objects

Action object:

accept_reject_order(order_query_no, client_no, accepted_rejected_status)

Uncertainty objects:

order_requirements (order_query_no, order_description, product_name, quantity_required, order_due_date)

client (client_no, client_name, client_contact_no, client_address)

current_workload (date, ongoing_order_count, pending_order_count, take_on_capacity)

expected_profit (order_query_no, expected_cost, expected_profit)

Objective objects:

maximize_profit (order_query_no, profit_earned, total_expenditure, total_selling_price) maximize_customer_satisfaction (client_no, client_feedback)

Decision 2: Can customer be a potential client
Action: Add client to contact list
Uncertainties:

- Client

Objectives:

- Maximize Marketing
- Maximize Customer Base

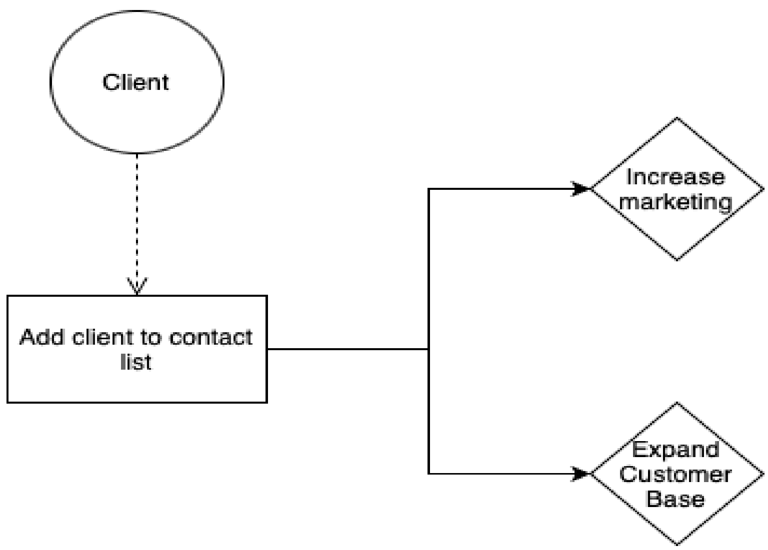

Fig 3.2 Influence Diagram for D2

Objects

Action object:

add_contact(contact_id, contact_name, contact_phone, contact_address)

Uncertainty objects:

contact_buisness(contact_id, contact_name, contact_address, company_name, company_description)

Objective objects:

maximize_marketing(contact_id, contact_name, company_name, contact_no, contact_address, contact_email)

maximize_customer_base(year, customer_count, new_customer_count)

Decision 3: Modify Production Plan or not
Action: Modify Production Plan
Uncertainties:

- Production Plan
- New Order
- Labour
- Raw Material

Objectives:

- Minimize Delay

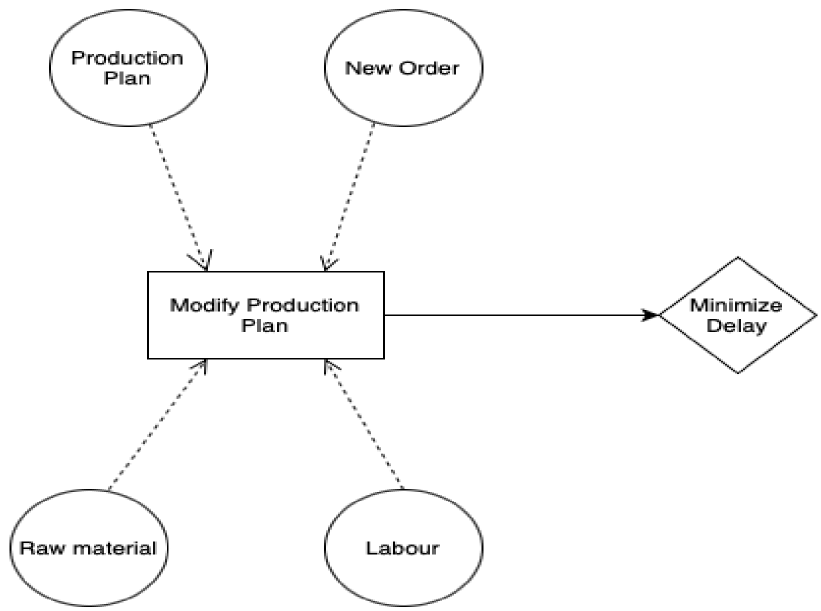

Fig 3.3 Influence Diagram for D3

Objects

Action object:

new_production_plan(month, week, order_id, order_schedule_task, complete_order_status, priority)

Uncertainty objects:

production_plan(month, week, order_id, order_schedule_task, complete_order_status)

new_order(order_id, product_id, quantity_required, order_recieved_date, order_due_date)

raw_material(month, week, item, expected_requirement, quantity_available)

labour(month, week, labour_requirement, labour_available)

Objective objects:

minimize_delay(order_no, product_no, order_due_date, order_completed_date, order_delivered_date)

Decision 4: Which order to prioritize
Action: Assign priority
Uncertainties:

- Customer
- Order

Objectives:

- Minimize Delay
- Maximize Customer Satisfaction

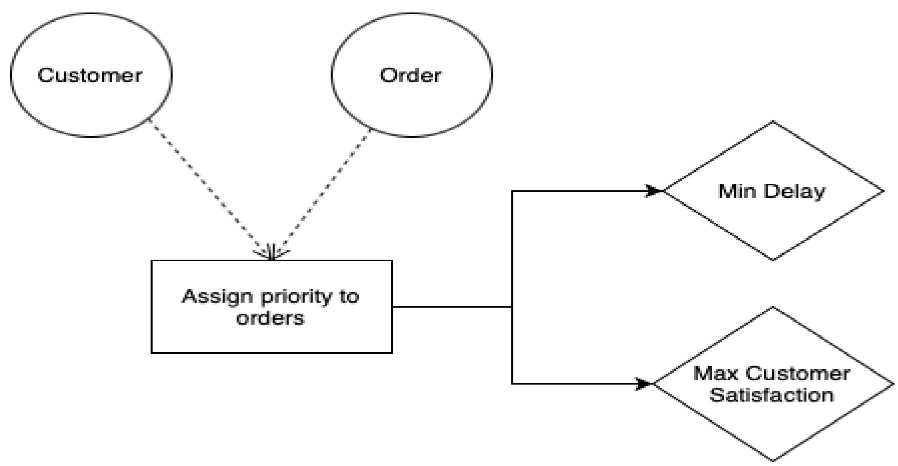

Fig 3.4 Influence Diagram for D4

Objects

Action object:

assign_priority(order_id,priority)

Uncertainty objects:

customer(customer_id, customer_name, customer_contact_no, customer_email)

order(order_id, product_id, quantity_required, order_recieved_date, order_due_date)

Objective objects:

minimize_delay(order_no, product_no, order_due_date, order_completed_date, order_delivered_date)

maximise_customer_satisfaction(customer_id, order_id, feedback)

Decision 5: Whether restock is required or not
Action: Stock check
Uncertainties:

- Current Stock

- Estimated Stock Requirement

Objectives:

- Minimize Delay
- Maximize Stock Availability

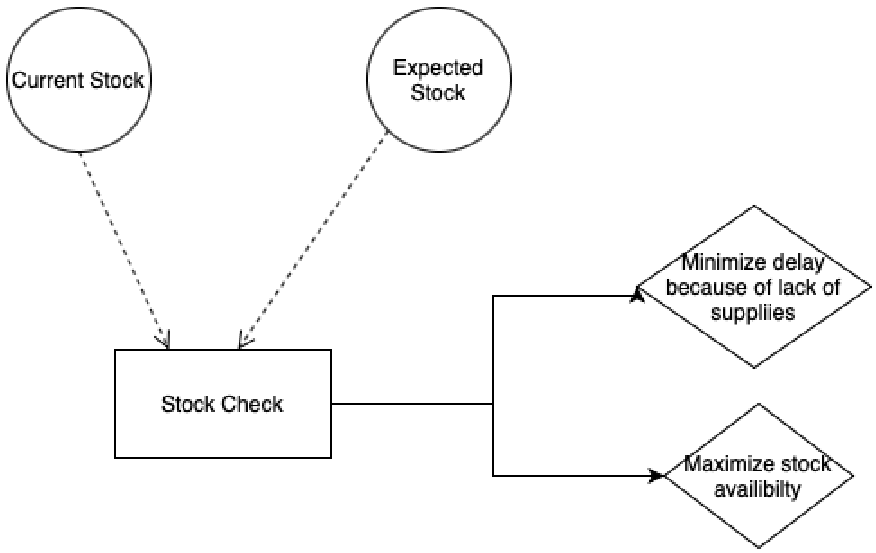

Fig 3.5 Influence Diagram for D5

Objects

Action object:

Stock_check (item_id, item_name, item_quantity,)

Uncertainty objects:

current_stock (item_id, item_name, item_quantity)

estimated_stock(order_no,product_id,item_id, expected_quantity)

Objective objects:

minimize_delay(order_no, item_id, quantity_required,quantity_available, order_due_date)

maximize_stock_availability(order_no, item_id, quantity_required,quantity_available)

Decision 6: When to restock
Action: Plan restock schedule
Uncertainties:

- Current Stock
- Estimated Stock Requirement
- Supplier

Objectives:

- Minimize Delay

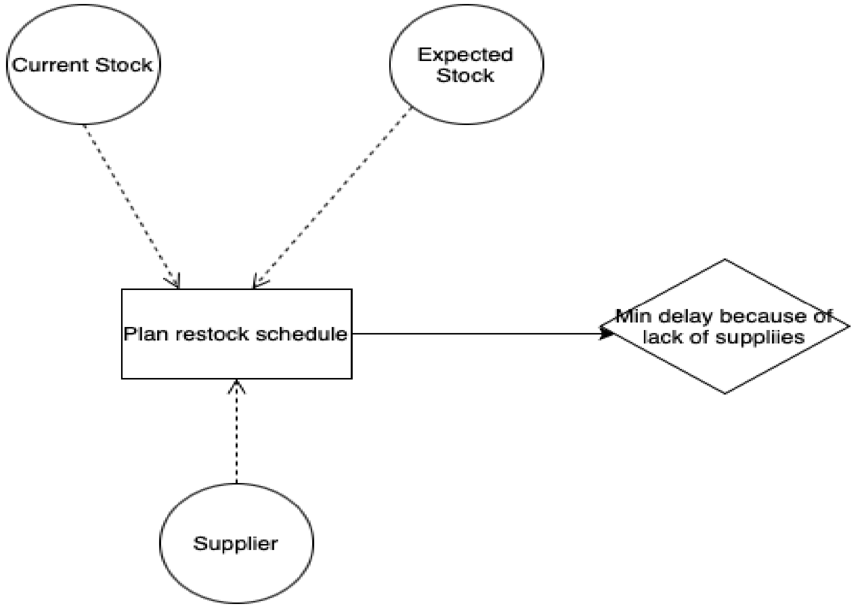

Fig 3.6 Influence Diagram for D6

Objects

Action object:

Plan_restock_schedule(item_id, item_quantity_required, restock_date)

Uncertainty objects:

current_stock (item_id, item_name, item_quantity)

estimated_stock(order_no,product_id,item_id, expected_quantity)

supplier(supplier_id, supplier_name, supplier_contact_no,supplier_address)

Objective objects:

minimize_delay(order_no,item_id, quantity_required,quantity_available, order_due_date)

Decision 7: Which items to restock
Action: Plan restock schedule
Uncertainties:

- Current Stock
- Estimated Stock Requirement
- Supplier

Objectives:

- Minimize Delay

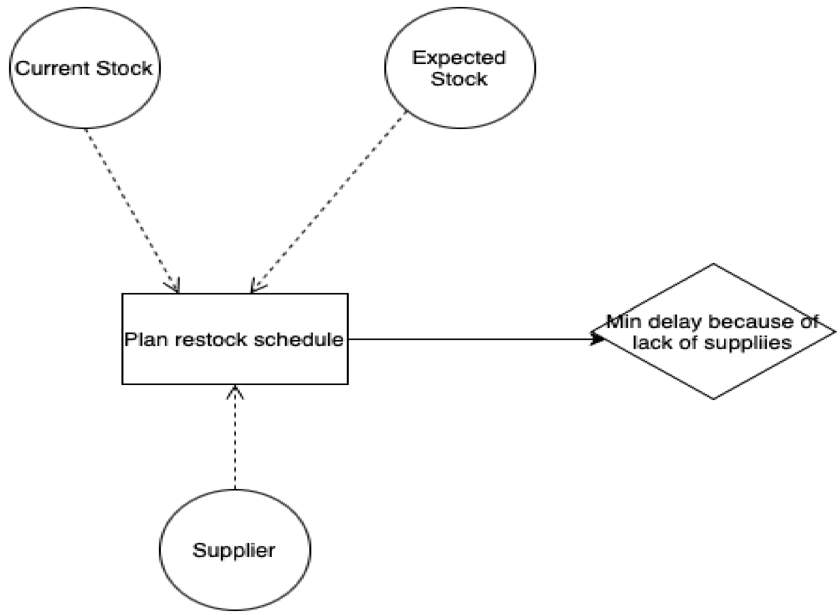

Fig 3.7 Influence Diagram for D7

Objects

Action object:

Plan_restock_requirements(item_id, item_quantity_estimate,item_price_per_unit, total_estimated_cost)

Uncertainty objects:

current_stock (item_id, item_name, item_quantity)

estimated_stock(order_no,product_id,item_id, expected_quantity)

supplier(supplier_id, supplier_name, supplier_contact_no,supplier_address)

Objective objects:

minimize_delay(order_no,item_id, quantity_required,quantity_available, order_due_date)

Decision 8: Which suppliers to buy supplies from
Action: Select supplier
Uncertainties:

- Supplier
- Materials Required

Objectives:

- Minimize Cost

- Maximize Quality

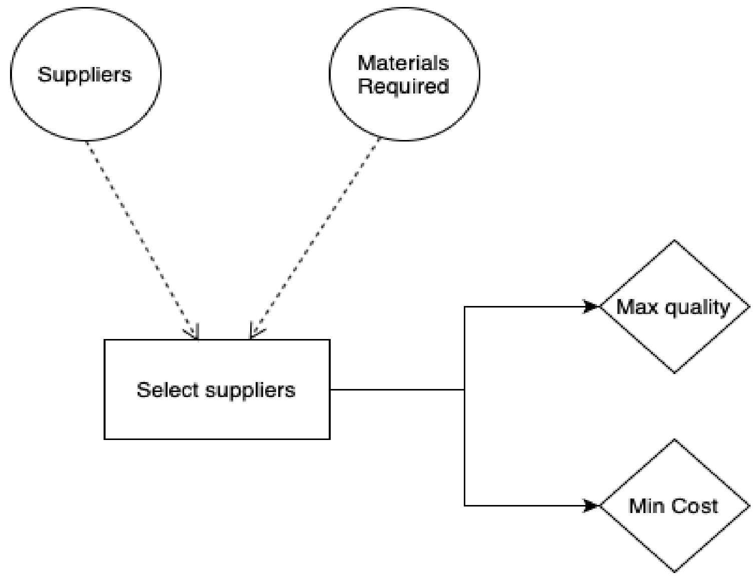

Fig 3.8 Influence Diagram for D8

Objects

Action object:

Select_supplier (item_id, supplier_id)

Uncertainty objects:

supplier(supplier_id, supplier_name, supplier_contact_no,supplier_address)

materials_required(item_id, quantity_required, time_available)

Objective objects:

minimize_cost(item_id, supplier_id, cost)

maximize_quality(item_id, supplier_id, past_experience)

Decision 9: How much quantity to be restocked

Action: Estimate quantity of materials required

Uncertainties:

- Current Stock
- Expected Requirement

Objectives:

- Min delay due to lack of supplies

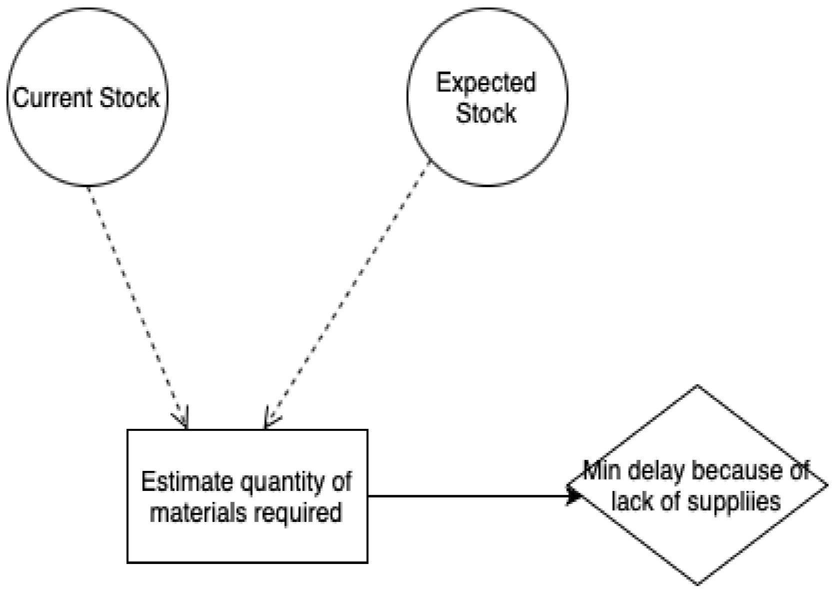

Fig 3.9 Influence Diagram for D9

Objects

Action object:

Plan_restock_quantity_requirements(item_id, item_quantity_required,item_price_per_unit, total_estimated_cost)

Uncertainty objects:

current_stock (item_id, item_name, item_quantity)

estimated_stock(order_no,product_id,item_id, expected_quantity)

Objective objects:

minimize_delay(order_no,item_id, quantity_required,quantity_available, order_due_date)

Decision 10: How many units to be produced daily

Action: Create daily production plan

Uncertainties:

- Production plan
- Expected Requirement

Objectives:

- Achieve daily target

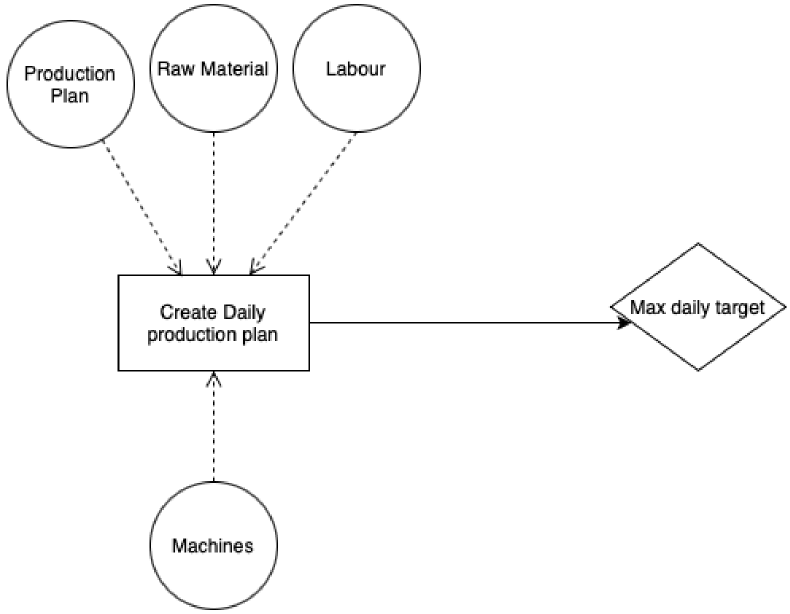

Fig 3.10 Influence Diagram for D10

Objects

Action object:

daily_production_plan(date, item_id, target_quantity, manufactured_quantity, remaining_quantity)

Uncertainty objects:

production_plan(month, week, order_id, order_schedule_task, complete_order_status)

raw_materials(date, item_id, quantity_available, quantity_required)

machine(machine_no, machine_status)

labour(date, labourers_required, labourers_available)

Objective objects:

maximize_daily_target(date, product_id, target_quantity, achieved_quantity)

Decision 11: How many units to be produced per machine
Action: Decide the machine-wise load.

Uncertainties:

- Daily production plan.
- Resources.

Objectives:

- Reach Daily Limit.

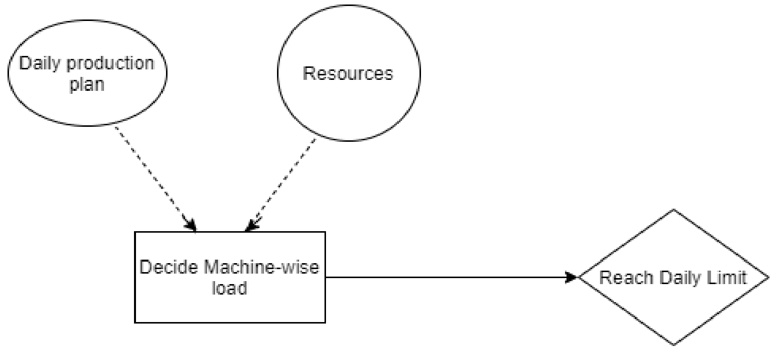

Fig 3.11 Influence Diagram for D11

Action Object:

- Decide_machine_wise_load(machine_number, machine_load, machine_status, date**)**

Uncertainty Objects:

- **Daily_production_plan**(date, product_id, target_quantity, manufactured_quantity)
- **Resources(** order_date, number_of_labours_available, number_of_machine_working, order_due_date**)**

Objective Objects:

- **Reach_daily_limit(**date, product_id, target_quantity, achieved_quantity)

Decision 12: Are more workers required

Action: Hire employees.

Uncertainties:

- Current workload.

Objectives:

- Maximize employee satisfaction.

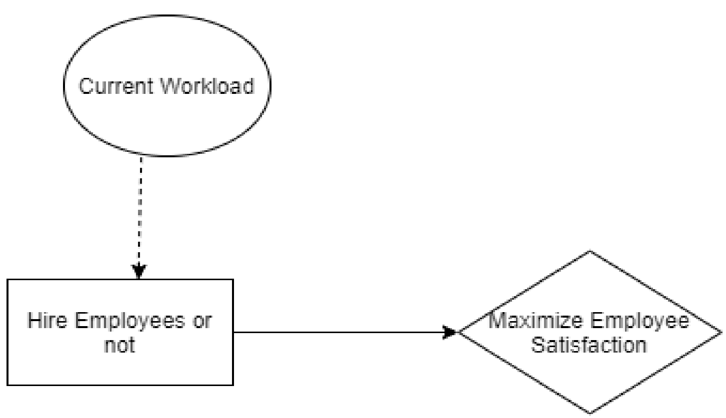

Fig 3.12 Influence Diagram for D12

Action Object:

- **Hire_employee(** potential_employee_id, employee_name, address, qualification, contact_number**)**

Uncertainty Objects:

- **Current_workload(** number_of_pending_orders, number_of_employees_available, numer_of_employees_required**)**

Objective Objects:

- **Maximize_employee_satisfaction(** employee_id, year, feedback**)**

Decision 13: What to do with poor quality products

Action: Handle poor quality products.

Uncertainties:

- Defective Products

Objectives:

- Minimize waste.

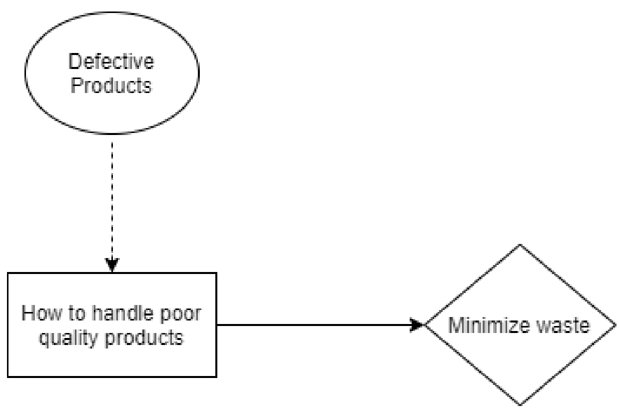

Fig 3.13 Influence Diagram for D13

Action Object:

- **Handle_poor_quality_products(** order_id, item_id, number_of_defective_products, cause_of_defect, action_taken **)**

Uncertainty Objects:

- **Defective_product(** order_id, product_id, product_type, number_of_defected_product **)**

Objective Objects:

- **Minimize_waste(** order_id, product_id, number_of_defected_products**)**

Decision 14: When to transport the products

Action: Transport products.

Uncertainties:

- Transporter
- Order Delivery

Objectives:

- Minimize cost.
- Minimize Delay.

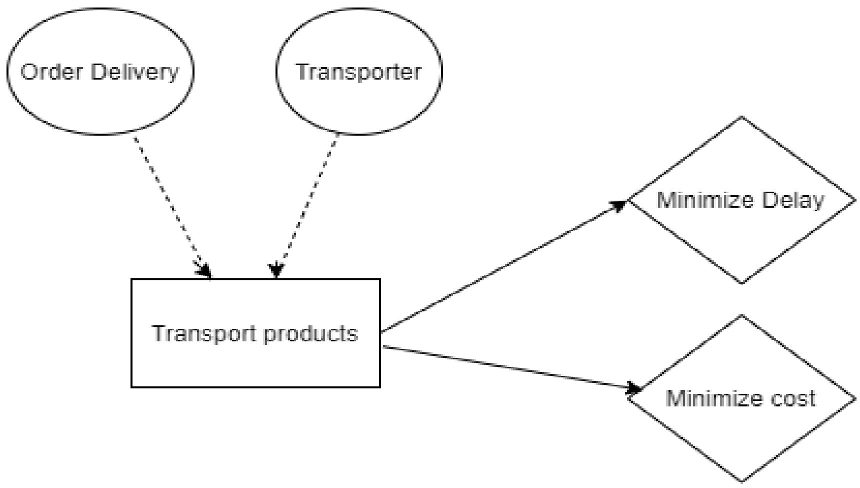

Fig 3.14 Influence Diagram for D14

Action Object:

- **Transport_products(**order_id, item_id, date_of_delivery, means_of_transportation, contact_information**)**

Uncertainty Objects:

- **Transporter (** transported_id, transporter_name, contact_information **)**
- **Order_delivery(** order_id, product_id, location, order_due_date**)**

Objective Objects:

- **Minimize_delivery_cost(**delivery_cost, product_id, order_id, means_of_transporation, location**)**
- **Minimize_delay(**order_id, product_id, manufacturing_date, dispatch_date, delivery_date, order_due_date**)**

Decision 15: Which mode of transportation should be used to transport the products

Action: Select a transportation mode.

Uncertainties:

- Transporter

- Order Delivery

Objectives:

- Minimize cost.
- Minimize Delay

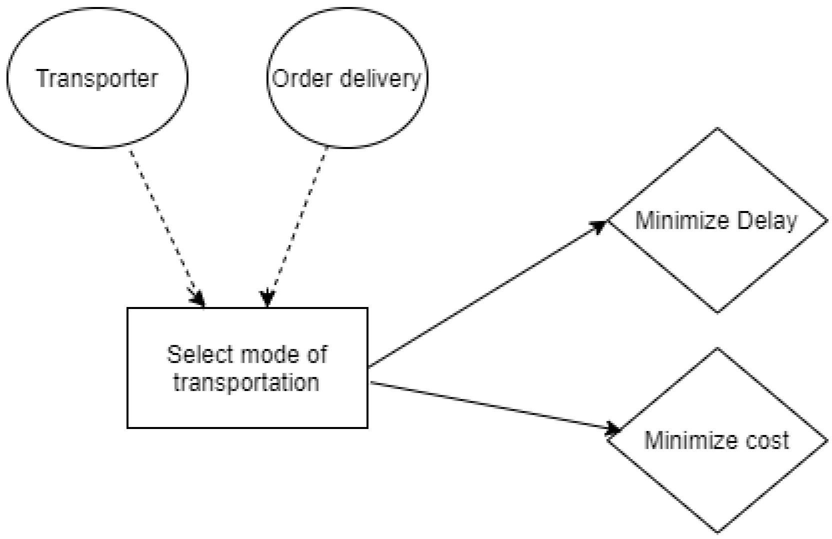

Fig 3.15 Influence Diagram for D15

Action Object:

- **Select_mode_of_transportation(**order_id, item_id, date_of_delivery, means_of_transportation, contact_information**)**

Uncertainty Objects:

- **Transporter(** transported_id, transporter_name, contact_information **)**
- **Order_delivery(** order_id, product_id, location, order_due_date**)**

Objective Objects:

- **Minimize_cost(** delivery_cost, product_id, order_id, means_of-transporation, location**)**
- **Minimize_delay(** order_id, product_id, manufacturing_date, dispatch_date, delivery_date, order_due_date**)**

Decision 16: When to make marketing calls

Action: Make calls.

Uncertainties:

- Work Status
- Contact business.

Objectives:

- Maximize marketing.
- Maximize customer base

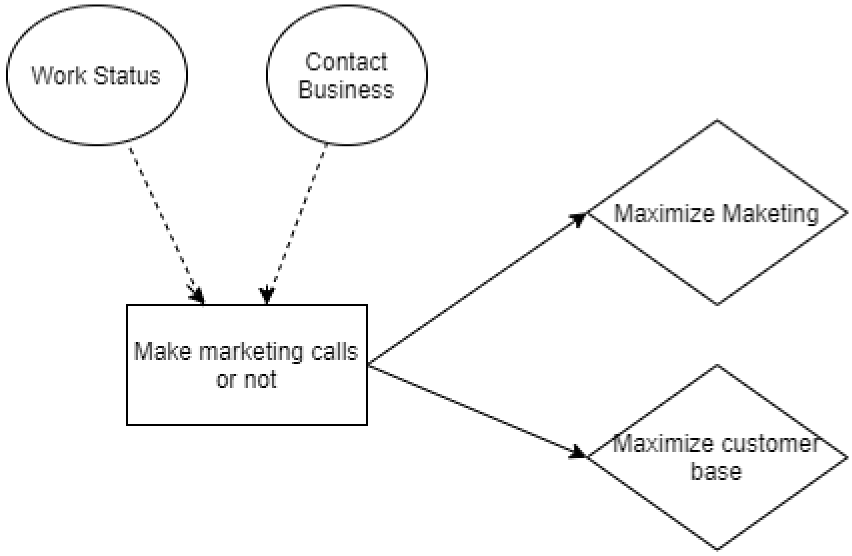

Fig 3.16 Influence Diagram for D16

Action Object:

- **Make_marketing_call(**contact_id, last_contact_date, call_feedback**)**

Uncertainty Objects:

- **Contact_buisness**(contact_id, contact_name, contact_address, company_name, company_description)
- **Work_status(** month, number_of_orders_receieved, number_of_calls_made, number_of_meetings_through_call**)**

Objective Objects:

- **Maximize_marketing**(contact_id, contact_name, company_name, contact_no, contact_address, contact_email)
- **Maximise_customer_base**(year, customer_count, new_customer_count)

Decision 17: When to service a machine

Action: Service machine.

Uncertainties:

- Machine Supplier
- Machine-wise load

Objectives:

- Minimize breakdown.

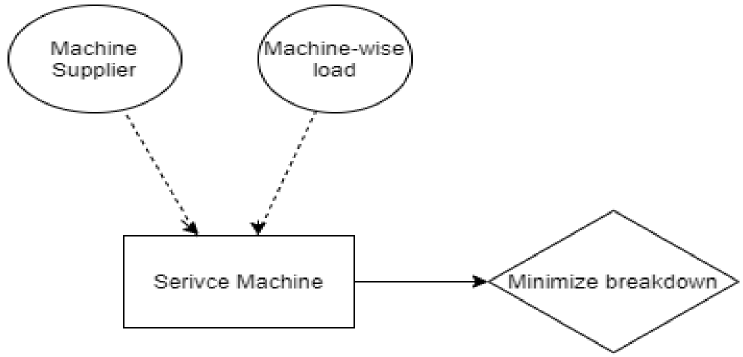

Fig 3.17 Influence Diagram for D17

Action Object:

- **Service_machine(**machine_number, fault_type, number_of_days_to_repair**)**

Uncertainty Objects:

- **Machine_supplier(**supplier_id , supplier_name, address**)**
- **Machine_wise_load(**date, number_of_machines. production_load**)**

Objective Objects:

- **Minimize_breakdown(** month, machine_id, servicing_cost, number_of_breakdowns, servicing_date**)**

Decision 18: Approve leave of an employee or not

Action: Approve leave.

Uncertainties:

- Employee.

Objectives:

- Maximize employee satisfaction.
- Minimize work disruption.

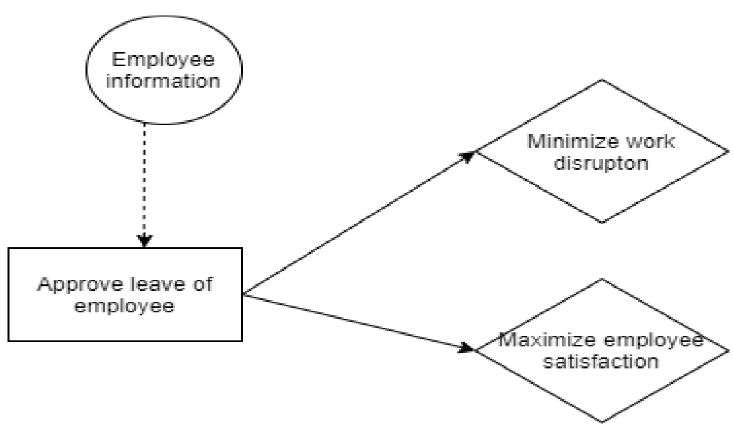

Fig 3.18 Influence Diagram for D18

Action Object:

- **Approve_leave(**employee_id, number_of_leaves_requested, date_of_leave, leave_grant_status, reason_for_leave**)**

Uncertainty Objects:

- **Employee_information(** emp_id, emp_name, contact, month, leave_record**)**

Objective Objects:

- **Maximize_employee_satisfaction(** employee_id, year, feedback**)**
- **Minimize_work_disruption (**month, number_of_orders, number_of_labourers_required, number_of_labours_available**)**

Decision 19: How many units to pack per box

Action: Pack box.

Uncertainties:

- Transporter
- Product packing

Objectives:

- Minimize cost.
- Maximize customer satisfaction.

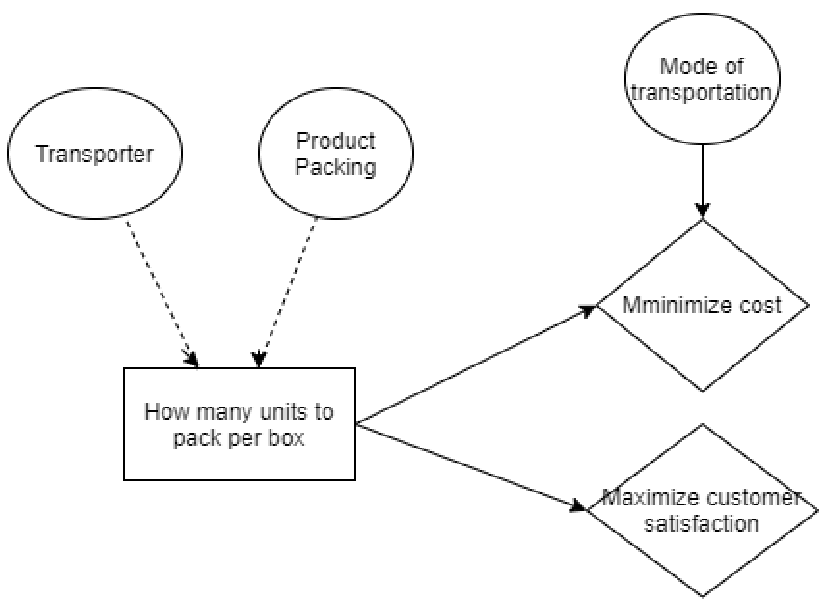

Fig 3.19 Influence Diagram for D19

Action Object:

- **Pack_boxes(**order_id, item_id, quantity_of_products, number_of_boxes_required, mode_of_transportation**)**

Uncertainty Objects:

- **Product_packing(** order_id, item_id, quantity**)**
- **Transporter(** transported_id, transporter_name, contact_information **)**

Objective Objects:

- **Minimize_cost(** order_id, number_of_boxes_required, cost**)**
- **Maximise_customer_satisfaction**(client_no, client_feedback)

3.4 DIEM tool for Sigma Enterprise

A tool has been implemented which captures different components of DIEM. The working of tool is as follows:

1) The home page consists of a navigation sidebar which contains three buttons home, DIEM tool and Backlog tool respectively. By clicking on the DIEM tool a user can enter inside a new web page where the DIEM tool is implemented.

Fig 3.20 Home page

2) This new web page also has a side navigation bar which contains different buttons and clicking each button redirects to the corresponding page where the user can enter details like decisions, actions, uncertainties, objectives and their attributes. The side bar also contains an option to update or delete the values entered in the tool.

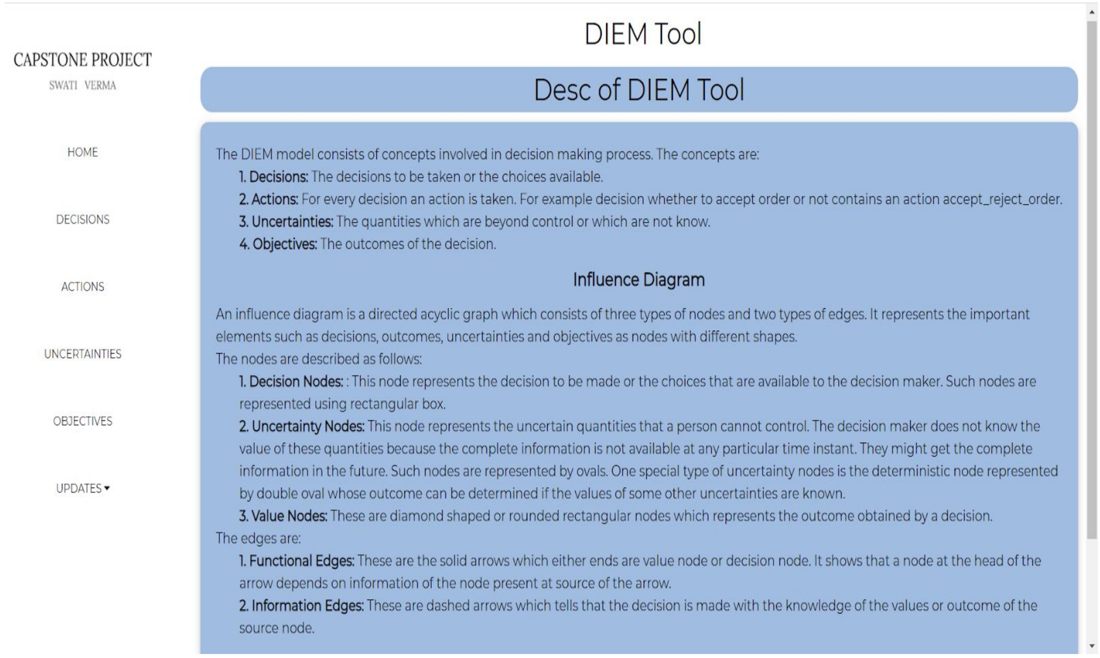

Fig 3.21 Home page of DIEM Tool

3) Following are the main pages of the tool:

i. Decision page: In this page the user can store the name of the decisions.
Here, I am showing all the components for the decision "Which Items to Restock?".

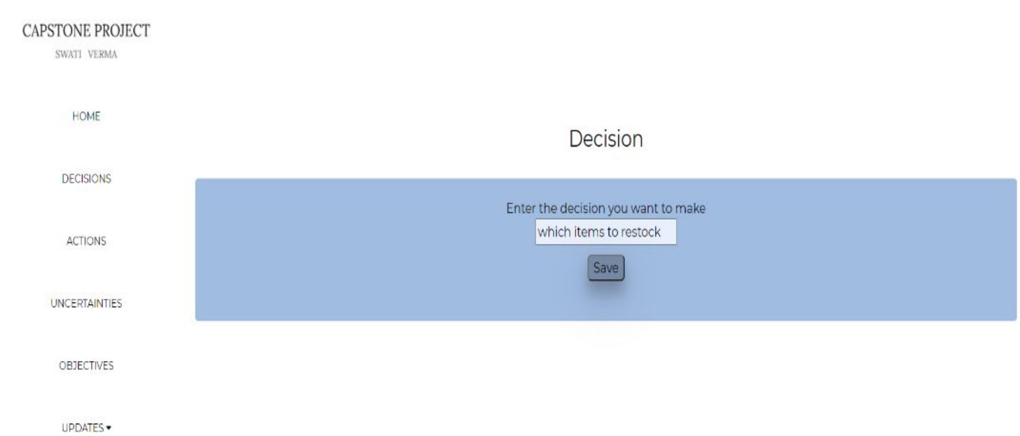

Fig 3.22 Decision Page

ii. Action page: In this page the user can store the actions corresponding to a decision. As soon as the save button is pressed, a new page, action attribute page is opened.
The action corresponding to which items to restock is Plan restock requirements which is stored below.

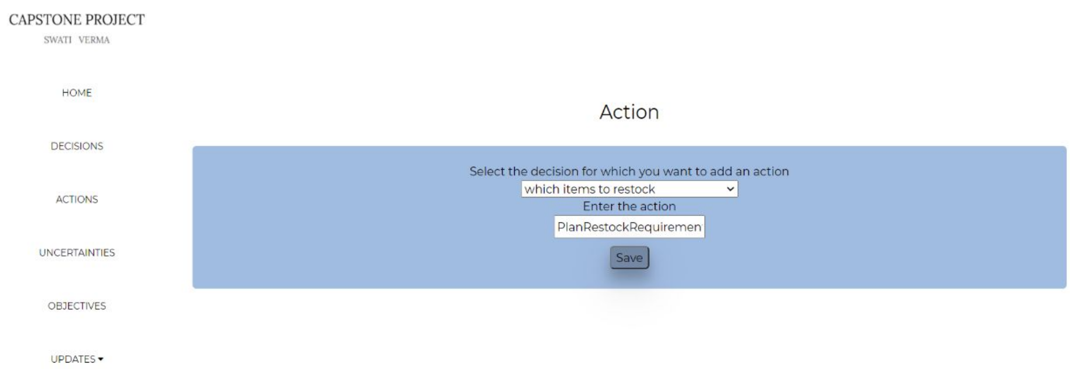

Fig 3.23 Action page

iii. Action attribute page: After entering the action name, the user needs to store the attributes and their data types corresponding to the action. Now there can be two types of attributes: base and derived. Derived attributes are the ones which can be derived from other attributes. So, in this page the type of the attributes are also stored and if there are any derived attributes then after clicking the save button a new web page base derived action attribute dependency will be displayed to enter which derived attributes are depending on which other attributes.

Attributes corresponding to plan restock requirements are shown below:

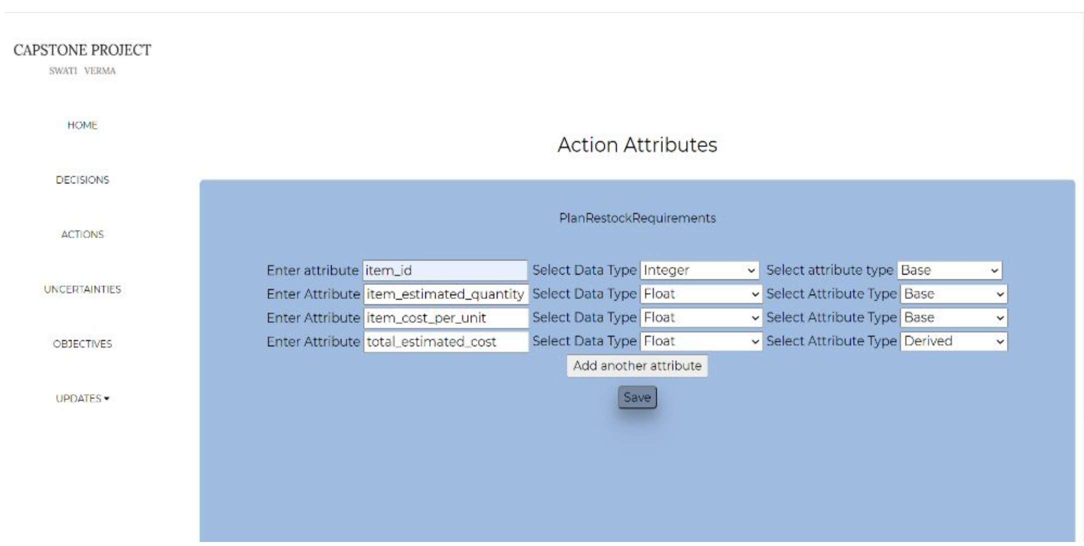

Fig 3.24 Action Attributes page

iv. Base derived action attribute page: This page captures the dependency between base and derived attributes. Plan restock requirements has one derived attribute whose dependency is stored in the below page.

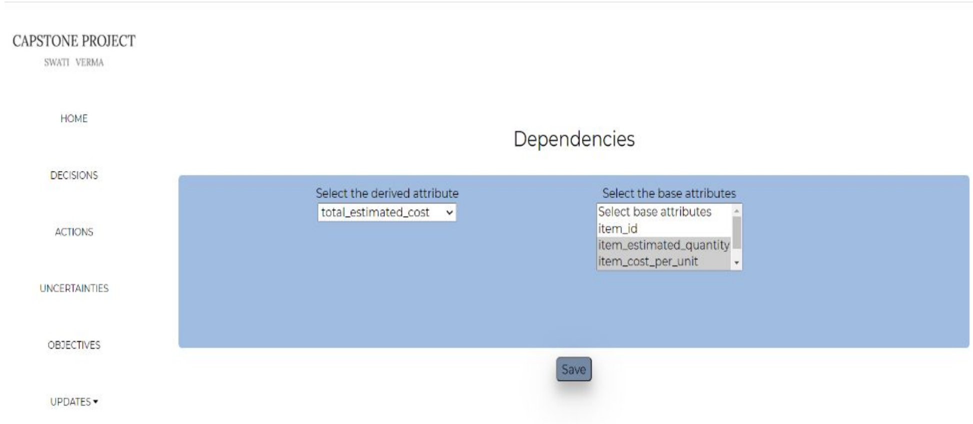

Fig 3.25 Action Attributes dependency page

v. Uncertainty page: In this page uncertainties corresponding to a decision are stored and after clicking the save button, a new page will be displayed which stores the uncertainty attributes. One of the uncertainties for the above action is current stock which is stored below.

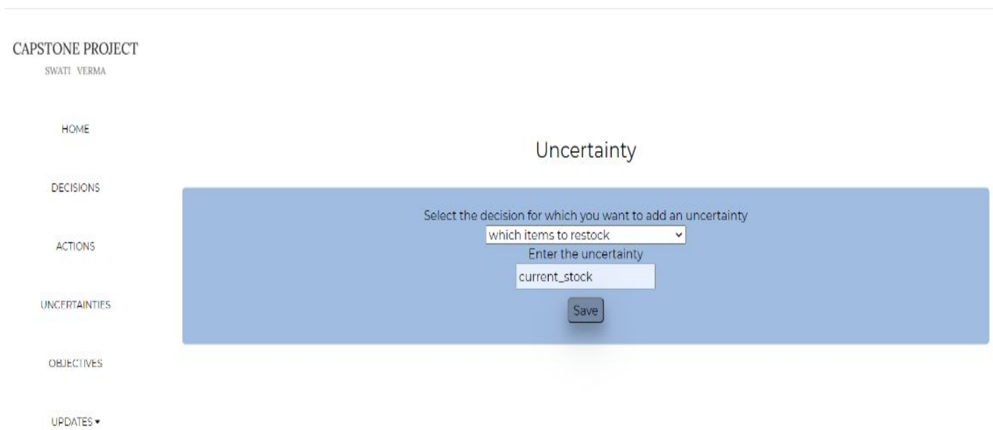

Fig 3.26 Uncertainty page

vi. Uncertainty attribute page: This page stores the uncertainty attributes and their data types.

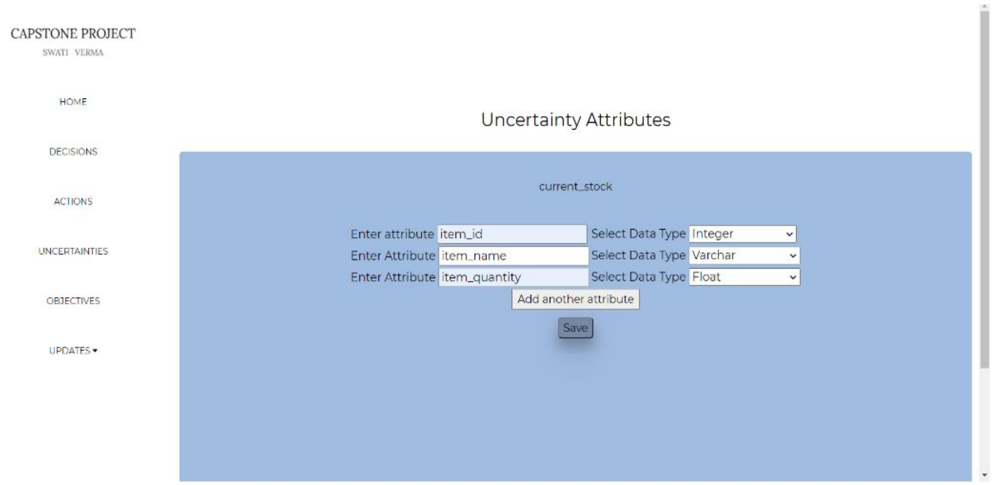

Fig 3.27 Uncertainty Attributes page

vii. Objective page: This page stores the objective corresponding to a decision and as soon as objectives are saved, a new page named objective attribute page is displayed. The objective for the above action is to minimize delay which is stored below.

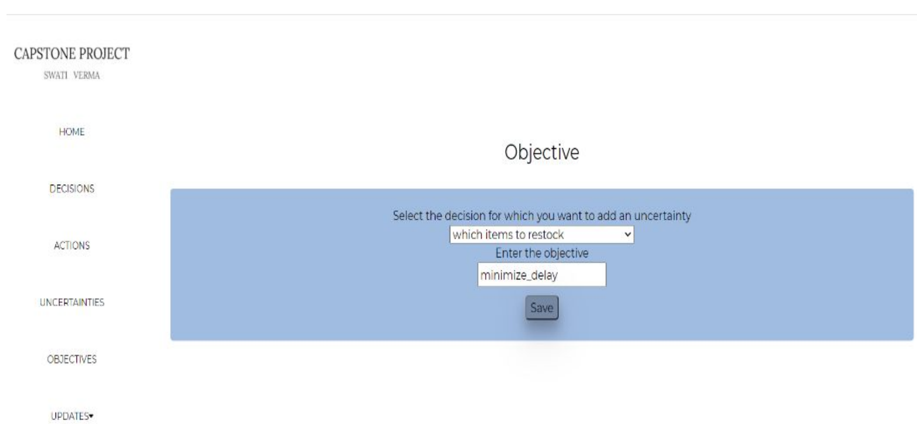

Fig 3.28 Objective page

viii. Objective Attribute page: This page stores the attributes and datatypes of an objective. This page also stores which attributes will be used for computation of the outcome. This information will be used in the choice phase of the information backlog tool.

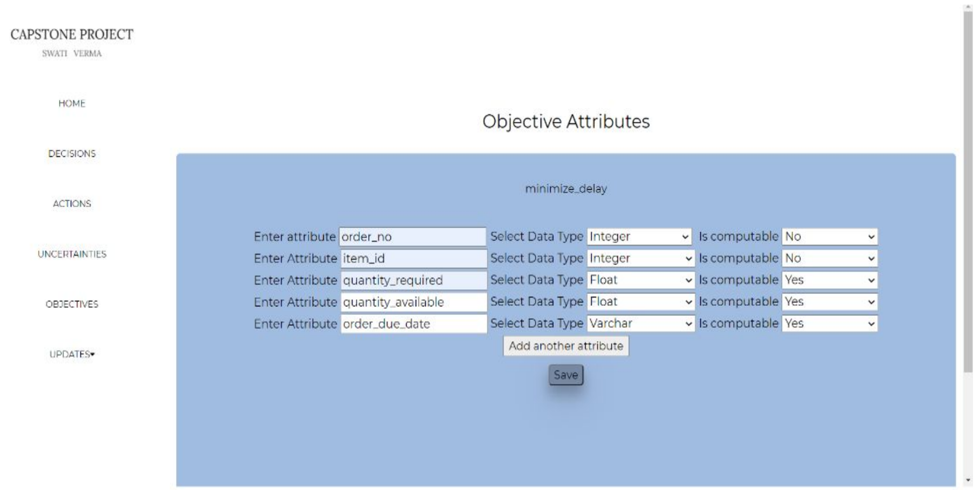

Fig 3.29 Objective Attributes page

ix. Update page: This page contains options for updation and deletion of the uncertainty names, action names and decision names. Update page for decision is shown below.

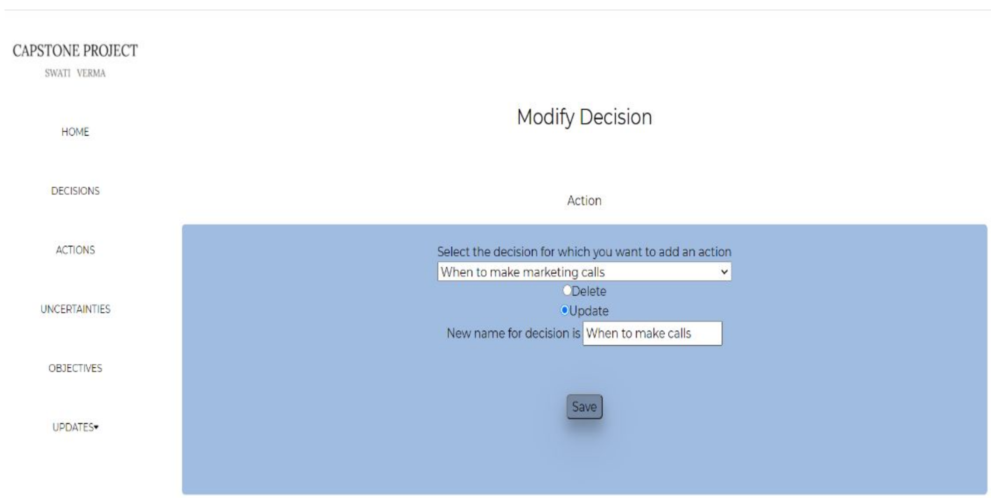

Fig 3.30 Modify Decision Page

4. Decision Backlog

4.1 Backlog of Decisions

The backlog here can be defined as the universe of all the decisions of interest. In this project four types of backlogs are implemented.[1]

4.1.1 Free Backlog: Standalone decision which does not depend on any other decisions, does not give rise to any specialized hierarchy and cannot be further decomposed to smaller decisions are considered are free decisions and such decisions belong to free backlog.

4.1.2 Specialized Backlog: This backlog consists of a set of atomic decisions which does not depend on any other decision and gives rise to a specialized hierarchy. For example, the root of the specialized hierarchy can be whether to buy a car or not. The children for this decision can be whether to buy an SUV, or to buy Wagon-R. Further, children can be whether the car should be a four seater or six seater, etc.

4.1.3 Composite Backlog: This backlog consists of decisions which are composed of smaller decisions and does not give rise to any specialized hierarchy and does not depend on any other decision.

4.1.4 Mixed Backlog: This type of backlog can contain composite, specialized or free decisions or the decisions which depend on any other decision.

A decision can be complete or incomplete. A complete decision can be defined as the decision

a. Which does not depend on any other decision or if it depends on decision X, then X should be taken before decision D is taken.
b. It should not be decomposed into smaller decisions or if it is composed of smaller decisions then all the smaller decisions must be taken before decision D is taken.
c. It should not be specialized or if it is specialized then all the decisions below in the hierarchy should be taken before decision D is taken.

For early development of the data warehouse, the standalone complete decisions are the best choice to take. So, the first preference is given to such decisions. The second preference is given to the specialized decision. Each node in the hierarchy is itself a complete decision. The root decision of the hierarchy is selected only after all the specialized decisions are taken. Third preference is given to the decisions belonging to the composite backlog.[1] Here also, the root of the composite decision is supported when all the smaller component decisions are already taken. At the end, least preference is given to the decisions of mixed backlog because of their complexity.

4.2 Selecting Decisions from Free Backlog

Initially, semantic priority is taken for all the decisions belonging to the free backlog. These priorities are given by the stakeholder. As all the decisions in this backlog are atomic and do not contain any other decisions so the structural priority for each decision here is the same. The highest priority decision D is taken first and is removed from the backlog. The decisions which are dependent on the D are now free and are added into the backlog. This process is repeated till the backlog becomes empty. [1] To explain the selection process, let us consider the following dependency graph:

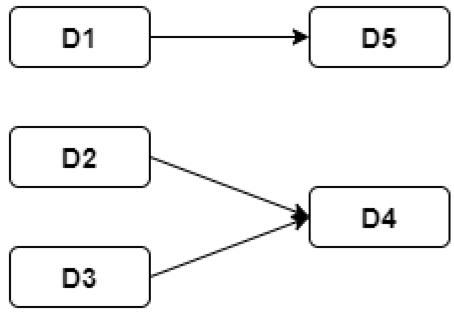

Fig 4.1 Free Backlog Decision Graph

The initial free backlog will contain {D1, D2, D3}. Let the semantic priorities given by the stakeholder are Sem(D1) = 0.5, Sem(D2) = 0.3 and Sem(D3) = 0.2. The priorities of all the decisions must sum up to unity. As the semantic priority of D1 is highest so the next selected decision will be D1 and it is removed from the backlog. Now, D5 is available for inclusion in free backlog, now the new backlog will be {D2, D3, D5}. Again, semantic priorities are taken for D2, D3 and D5. Let the next decision selected be D2, it is removed and the new free backlog = {D3, D5}. Now let the next decision selected be D3 and it is removed from the backlog. Now, D4 is also available for inclusion and is included in the backlog. New backlog = {D4, D5}. Again, semantic priority is taken from stakeholders and the process continues.

4.3 Selecting Decisions from Specialized Backlog

Specialized hierarchy consists of a set of decisions which gives rise to a specialized hierarchy. Here, we give preference to the syntactic priority of the decisions than semantic priority. Syntactic priority or Ssyn is calculated as the number of its specialized decisions. We take an assumption here that the more the value of Ssyn, the more its lead time will be. So, the lower value of Ssyn is considered first.

The procedure for selecting decisions from specialized is described below:

1) Initially the decisions in the backlog are assigned semantic priorities which sums up to unity by stakeholders. The decisions are taken prioritize on the basis of syntactic priority. The node with the lowest value of syntactic priority is taken first. Let the first decision taken be D.

For the tree rooted at D, perform the following steps:

2) Move to the next lower level of the hierarchy. If there is no such level or if the node is leaf then go to step 4.
3) Choose the decision with least syntactic priority, Ssyn. If more than one decisions have same syntactic priority then look at their semantic priorities and
    i. Select the decision with highest semantic priority of that level.
    ii. If semantic priorities are also the same then select the next decision by interacting with stakeholders.

   Repeat above step till leaf is marked.
4) The selected leaf is then removed from the hierarchy.

5) If there are marked nodes still present in the hierarchy then move to next higher level and repeat from step 2.[1]

Let us take a specialized hierarchy to better understand the selection procedure.

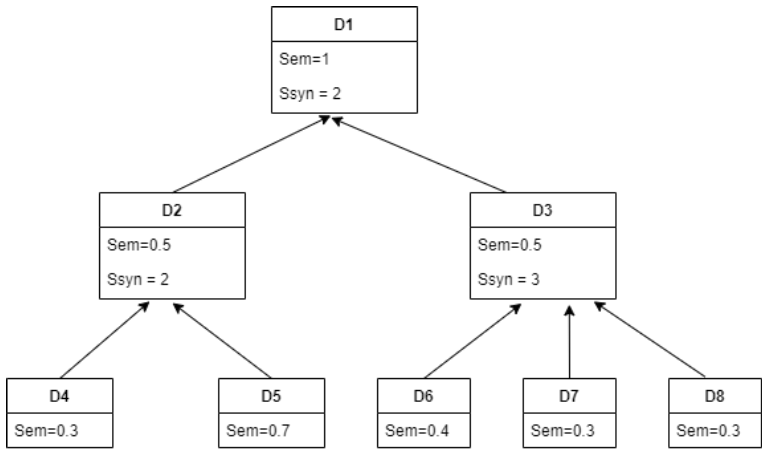

Fig 4.2 Specialized Decision Hierarchy

As the root of the hierarchy is D1, so initially D1 is marked and then next lower levels are traversed. In the next level, we have decisions D2 and D3. As the syntactic priority of D3 is greater than that of D2 so now D3 is marked and its lower level is searched. Now as all the nodes in the next level are leaf nodes. As the syntactic priority for D4 and D5 is the same, semantic priorities are seen and as D5 has higher semantic priority, so it is declared selected and is removed from the hierarchy. The next decisions taken are D4 and D2. Now the same procedure is followed for D3 until D1 is selected. The final order of the selected decisions will be {D5, D4, D2, D7, D8, D6, D3, D1}

4.4 Selecting Decisions from Composite Backlog

In the composite backlog, for a larger decision to be taken all its component decisions must be taken. While selecting composite decisions semantic priority is seen first and then syntactic priorities are seen.[1]

The procedure for selecting decisions from composite backlog is described below:

1)Initially the decision with highest semantic priority is selected.

2) Move to the next lower level of the hierarchy. If there is no such level or if the node is leaf then go to step 4.

3) Select a decision with the highest value of Sem. If more than one decisions have same semantic priority then calculate the syntactic priority or Csyn of the decision and

i. Select the decision with lowest Csyn.

ii. If multiple decisions have the same Csyn then resolve the conflict by interacting with stakeholders.

Repeat above step until leaf is reached.

1) The leaf node is selected and removed from the hierarchy.

2) move to the next higher level and repeat from step 2.

Let us take a composite hierarchy to better understand the selection procedure.

Initially the backlog contains decision D1. The next lower levels are seen. As D2 and D3 have the same semantic priorities, syntactic priorities are seen and D2 is selected as it has lower value of syntactic priority.

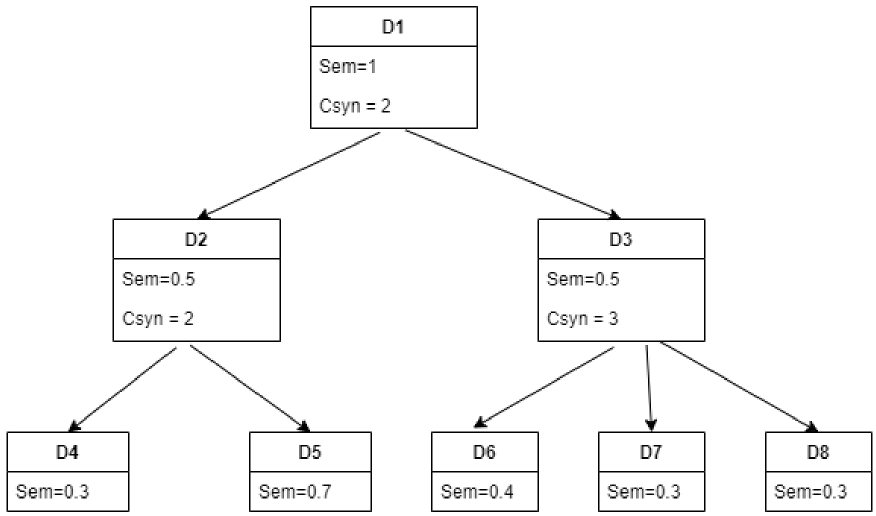

Fig 4.3 Composite Decision Hierarchy

Now in the next level D4 and D5 are potential decisions among which D5 is selected and removed as it has higher semantic priority. Next selected decisions are D4 and D2. Now D3 is still remaining so its children are seen and D6 is selected as its semantic priority is highest. Now D7 and D8 both have the same semantic priorities so now through interaction with stakeholders this conflict is resolved. Let the next selected decision be D7. So, the next selected decisions will be D8, D3 and D1. The final order will be {D5, D4, D2, D7, D8, D6, D3, D1}

4.5 Selecting Decisions from Mixed Backlog

Decisions are selected from a mixed backlog in the following manner.

1) For decisions which have Csyn=0 and Ssyn !=0, the procedure for selecting decisions from specialized backlog is followed.
2) For decisions which have Csyn !=0 and Ssyn =0, the procedure for selecting decisions from composite backlog is followed.
3) For decisions which have Csyn !=0 and Ssyn !=0, the procedure for selecting decisions from specialized backlog is followed after that selecting decisions from composite backlog is followed.

In this project, the dependency graph is captured which is a kind of mixed backlog only.

4.6 Selecting Decisions from Decision Backlog Tool for Sigma Enterprise

In chapter 1, all the types of decisions are identified for Sigma Enterprise. It is found that specialized decision is not present in our project but is implemented in the tool.

The tool consists of a home page for backlog tools. This page consists of a small description about the tool and has one side bar which redirects you to different pages.

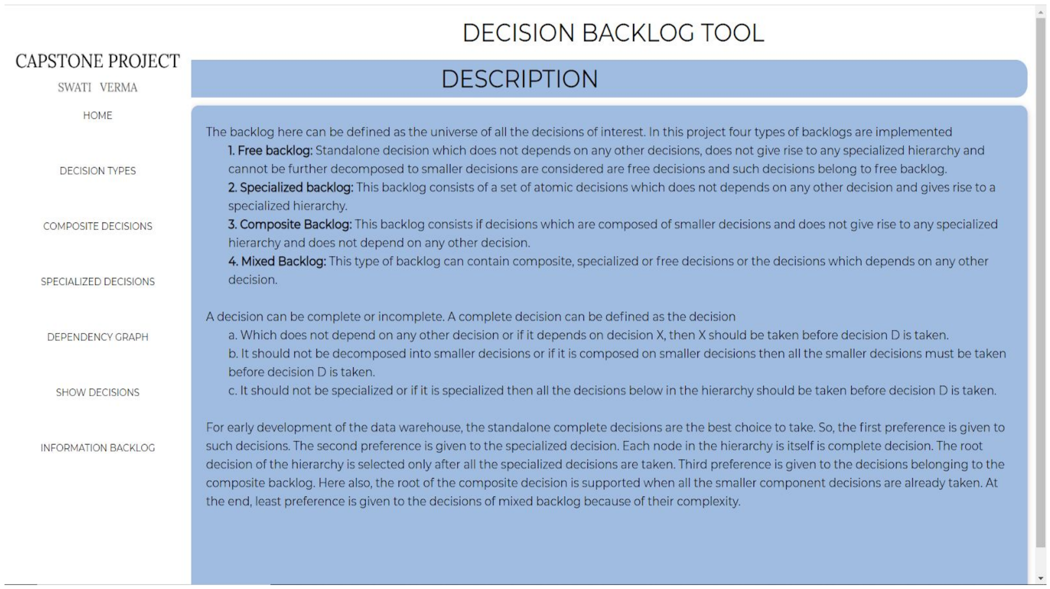

Fig 4.4 Home page for Decision Backlog

The main pages of this tool are:

1) Decision Types Page: This page shows all the decisions and captures the type of backlog to which that decision belongs. Three kinds of backlogs are considered namely free, composite and specialized. All the decisions of Sigma Enterprise and their types are shown in the below page.

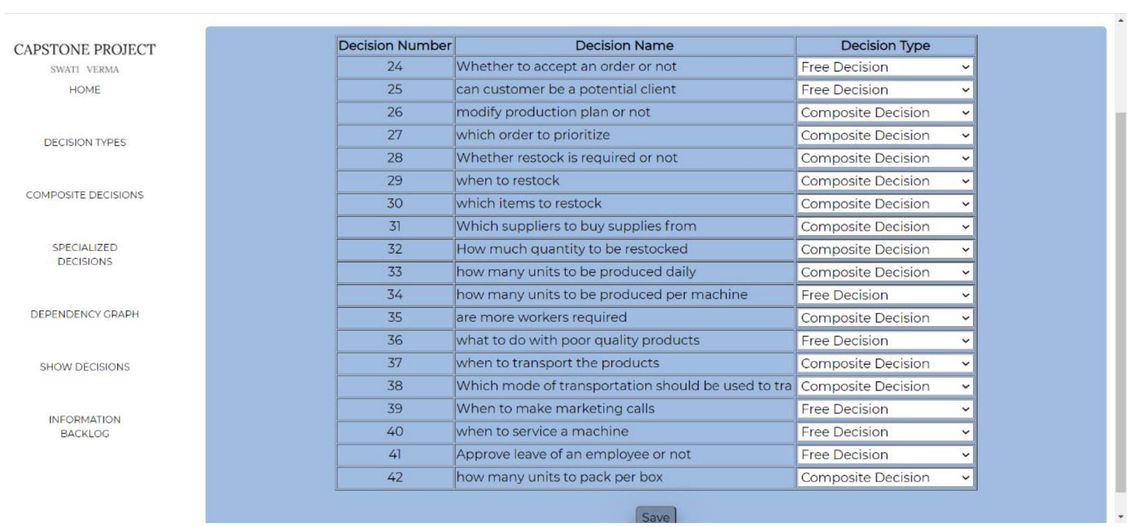

Fig 4.5 Decision Types Page

2) Composite Decision Page: This page captures the composite decision hierarchy like larger decisions composed of which smaller components. There are three composite decisions of Sigma enterprise. One such composite decision is whether restock is required or not which is composed of four decisions namely, when to restock, which items to restock, how much quantity to be restocked and which suppliers to buy supplies from. Such dependencies are captured using this page.

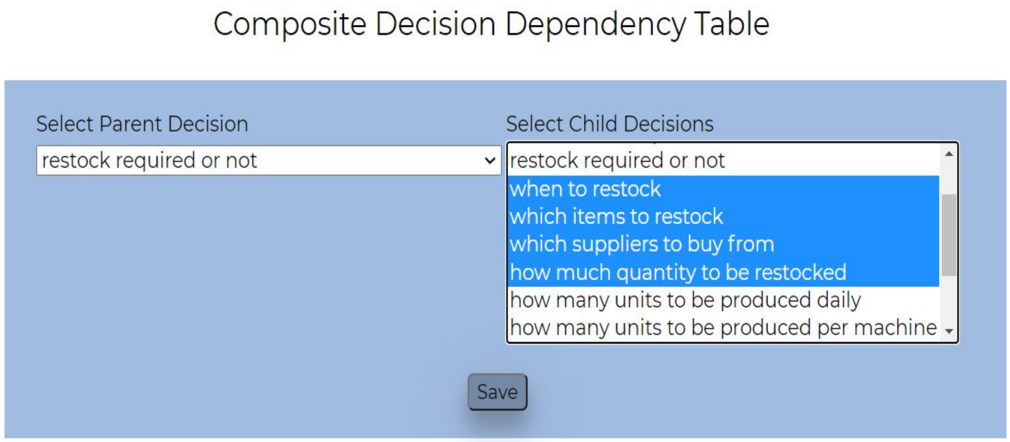

Fig 4.6 Composite Decision Page

3) Specialized Decision page: This page captures the specialized hierarchy. As Sigma enterprise has no specialized decision so this page has no information.

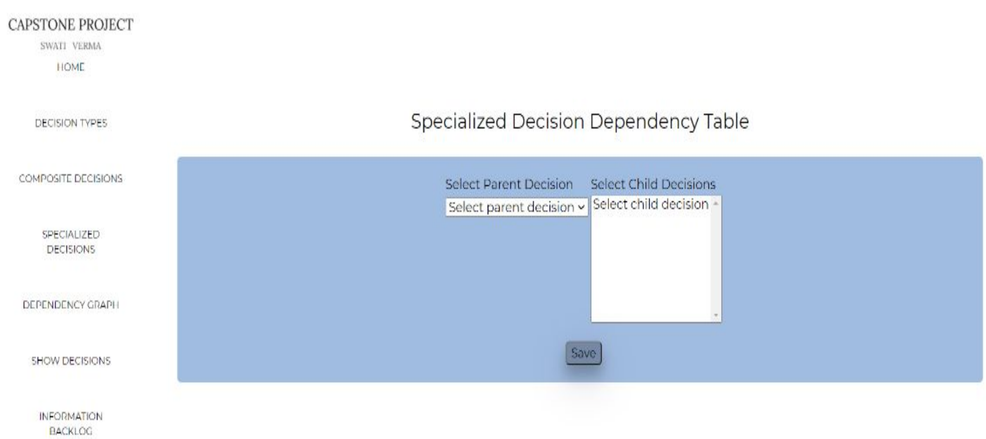

Fig 4.7 Specialized Decision Page

4) Dependency Graph Page: This page captures the dependency graph of the Sigma Enterprise created in chapter1. Using one dropdown menu you have to select the parent

decision and using other dropdown menu you can select the child decisions and by clicking at save button you can save them.

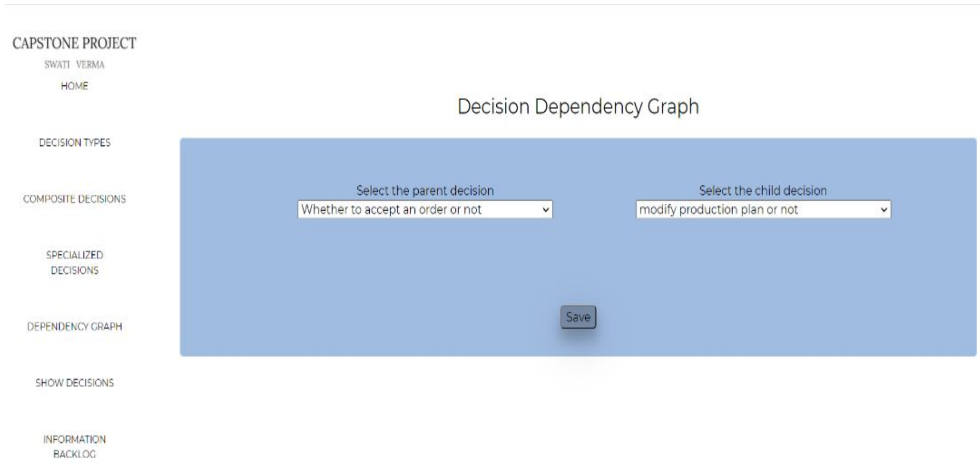

Fig 4.8 Decision Dependency Graph Page

5) Show Decision page: This page shows the order in which the decision should be selected from the backlog of decisions. Initially the semantic priorities summing to unity are taken for all the decisions which belong to a free backlog. Following are the decisions in the free backlog at the initial stage:

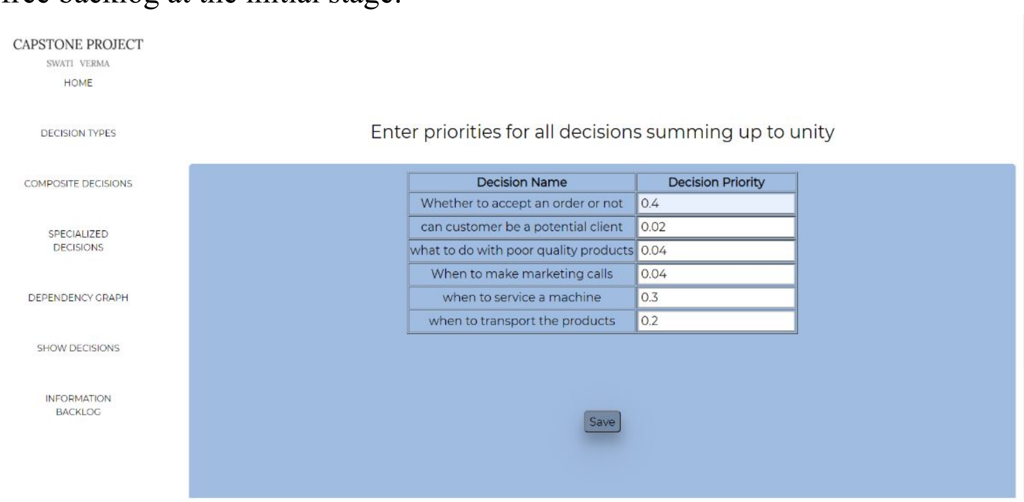

Fig 4.9 Get Initial priority Page

The highest semantic priority is given to the decision whether to accept order or not which is a free decision. So, the next decision selected must be accept order or not

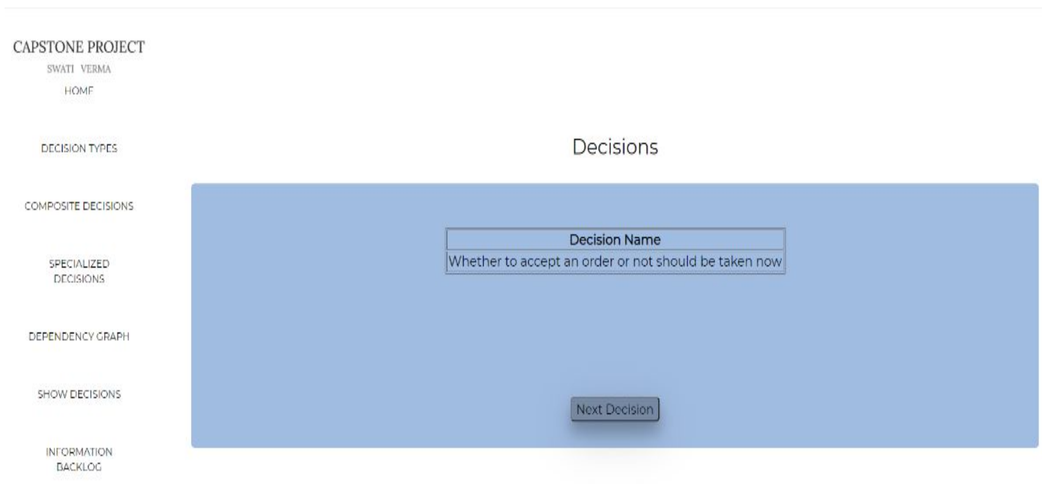

Fig 4.10 Show Decision Page

Again, according to the dependency graph after accepting order or not is selected and removed from the backlog, "modify production plan or not" will become free and will be included in the backlog. Now, again the new priorities are taken from the stakeholders.

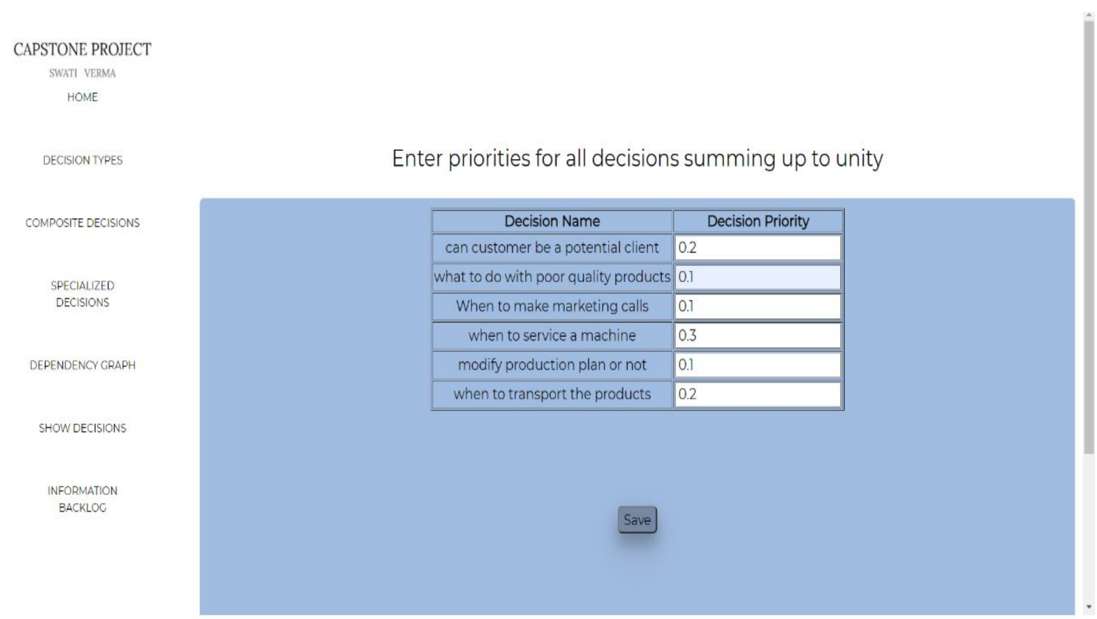

Fig 4.11 Get priority Page

Now as when to service a machine is given highest priority, it will be selected next.

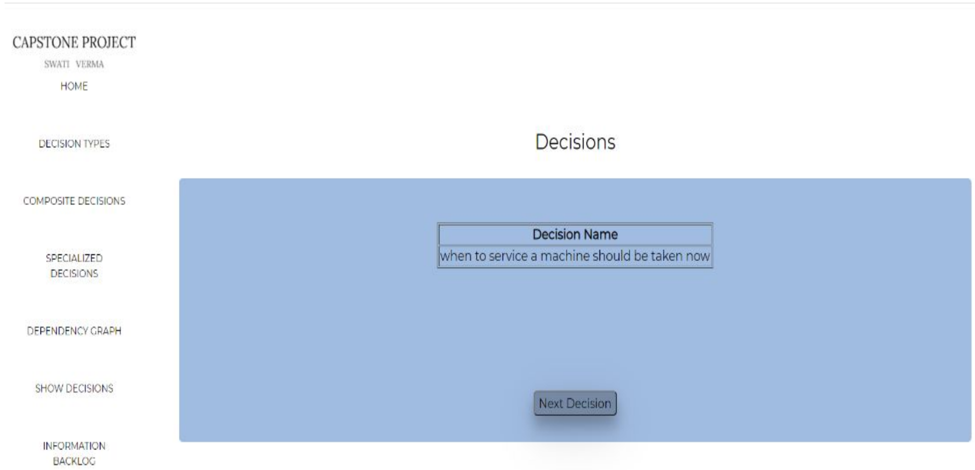

Fig 4.12 Show Decision Page

Now "can customer be a potential client" and "when to transport the products" are having same semantic priority, but the former is free decision and later is composite decision. So, the next decision selected should be can the customer be a potential client.

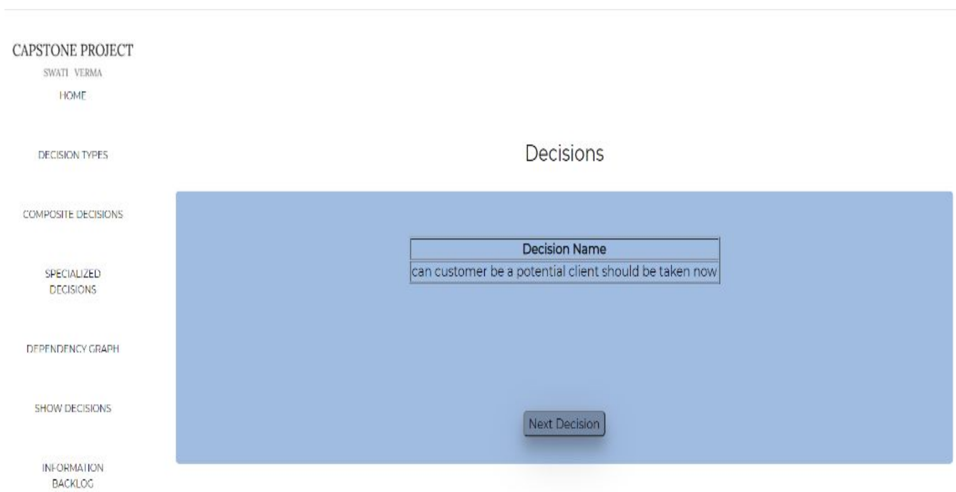

Fig 4.13 Show Decision Page

Similarly, all the free decisions are selected before composite or specialized decisions which are present in the current backlog.

When a composite decision comes then the priorities of its children is again asked by the stakeholders. For example, one of the composite decisions are "when to transport the products" so the priorities for the child are asked first and then all the nodes at the lower levels are selected first and then the root is selected at the end.

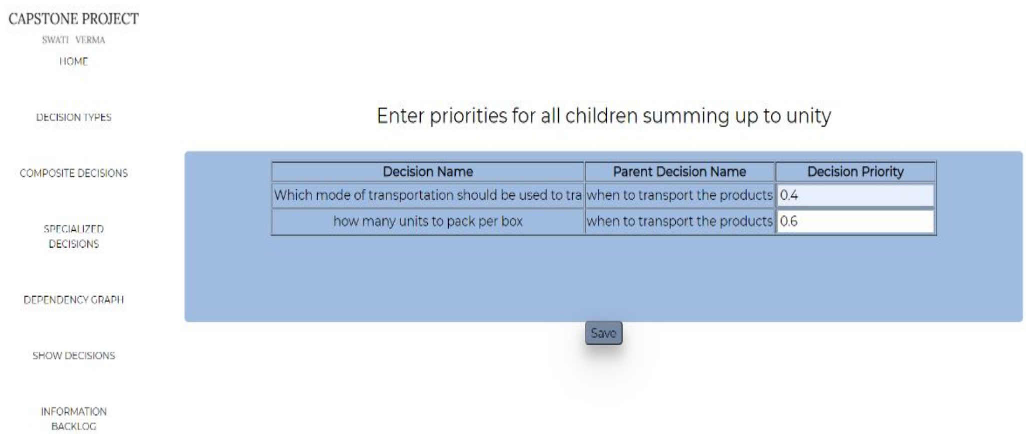

Fig 4.14 Get Child Priorities Page

If the semantic priorities of two or more decisions are the same then this conflict is resolved by asking stakeholder to choose one of the decisions. The below page shows one of such scenarios where the priorities of two decisions are 0.3 and 0.3 and the stakeholder has to select one decision to proceed further.

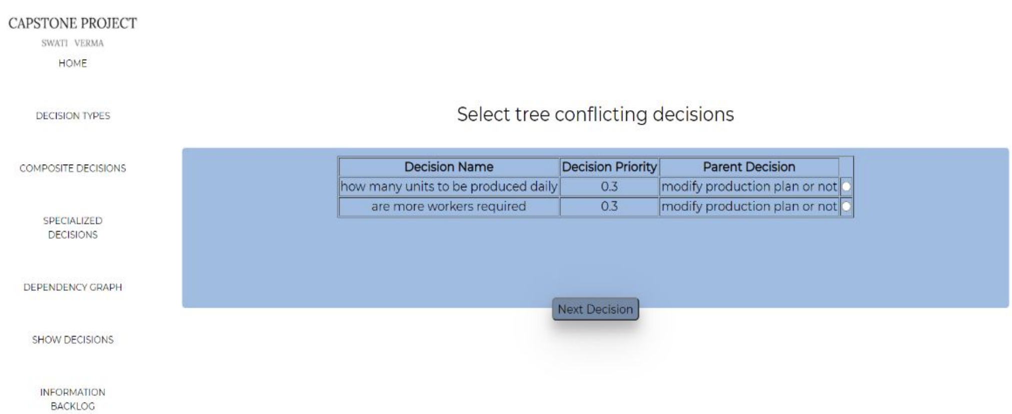

Fig 4.15 Conflicting Decisions Page

Similarly, all the decisions are shown one by one in the order in which they should be taken for early development of the data warehouse.

After all the decisions are shown the blank page is shown with no decision name. To see all the decisions obtained in the sequence you can go to the information backlog page shown in the sidebar. The working of Information backlog page is described in the next chapter.

5. Information Backlog

After the decisions are selected, the next step is to decide the information that has to be stored in the data warehouse. The comes in picture of the uncertainties, actions and outcomes of the influence diagram. The objective of the influence diagram is computed as a function of form F(x) = y. These x and y contribute to the information backlog. Here comes the intelligence and choice phase supported by the data warehouse. For the intelligence phase the actions corresponding to a decision is used to generate information because the sequence of past actions gives information about the current situation of the organization. For the choice phase, different alternatives are given and the user has to make a choice. This phase uses the uncertainties and outcomes of the influence diagram.

5.1 The Intelligence Phase

This phase contains the current situation of the enterprise. Relevant actions are considered for this phase. Let the action corresponding to a decision be ACT and let ACT be containing a record R(A1,A2, A3…An) where A1, A2…An are attributes of the action. The record R may be accessed by other decisions ACT1, ACT2…ACTp. So, the records of all such actions are now useful to get proper information. Again, the records of such actions will be accessed by other actions and the process continues till no new attributes or actions are found.[1]

The information backlog will be as follows:

- The record R (A1, A2,…An).
- All its base attributes BA1, BA2…BAm and derived attributes DA1,DA2…Dp.
  Repeat till no new actions or attributes are left

  i.The records of all the actions ACT1, ACT2,….ACTp that access base or derived attributes.

  ii. Attributes that are derived from new actions ACT1, ACT2…ACTp.

For example in Sigma enterprise, the decision "which items to restock contains the action and records as Plan_restock_requirement(item_id, item_quantity_estimate,item_price_per_unit, total_estimated_cost). Here the derived attribute is total_estimated_cost which depends on the item_quantity_estimated and item_price_per unit. So, now the information backlog would contain 1. Plan_restock_requirements and the derived attribute total_estimated_cost.

Now, the action corresponding to the decision how much quantity to be restocked is accessing the derived attribute total_estimated_cost and its action name is also plan_restock_requirements. So, now the information backlog will contain plan_restock_requirements along with the previous two information. This process is repeated till no new attributes or actions are remaining.

5.2 The Choice Phase

In this phase, all the outcomes of all the alternatives can be evaluated. If the information backlog is too large then we can select the subset of the alternatives or the set of outcomes.

Let the information backlog B for a decision D contain a set of outcomes O ={O1, O2,…On} and let B= {B1, B2,…Bm} be the subset such that outcome Oi is evaluated using Bi.[1]

The selection process for the choice phase is as follows:

1. Assign semantic priorities to the outcomes which sums up to unity.

REPEAT

2. Select the highest priority outcome and get the alternatives.
3. Estimate the information size for all the alternatives.
   i. If the size is too small then repeat from step 2 for the next higher priority outcome.
   ii. If the size is too large then break the information into a number of sprints and ask the user to select a subset of alternatives for the next sprint.
   iii. If the size is appropriate then select the information for the sprint.

UNTIL NO MORE OUTCOMES

The example for this phase is described along with the tool.

5.3 Information Backlog Tool for Sigma Enterprise

Initially all the decisions taken are shown in the order obtained during selecting decision from decision backlog and two radio buttons are given one for intelligence phase and other for choice phase. The user has to select either of the radio buttons and then click get info in order to get information of a particular phase.

Fig 5.1 Information Backlog Page

Here the user has selected the decision which items to restock and the intelligence phase shows the action name, its record and derived attributes. If the user will click on the more information button then the other actions which access the base or derived attribute of current action will be shown along with their records. Sigma enterprise contains only one derived attribute total_estimated_cost. The types of attributes whether they are base or derived are stored using the DIEM tool.

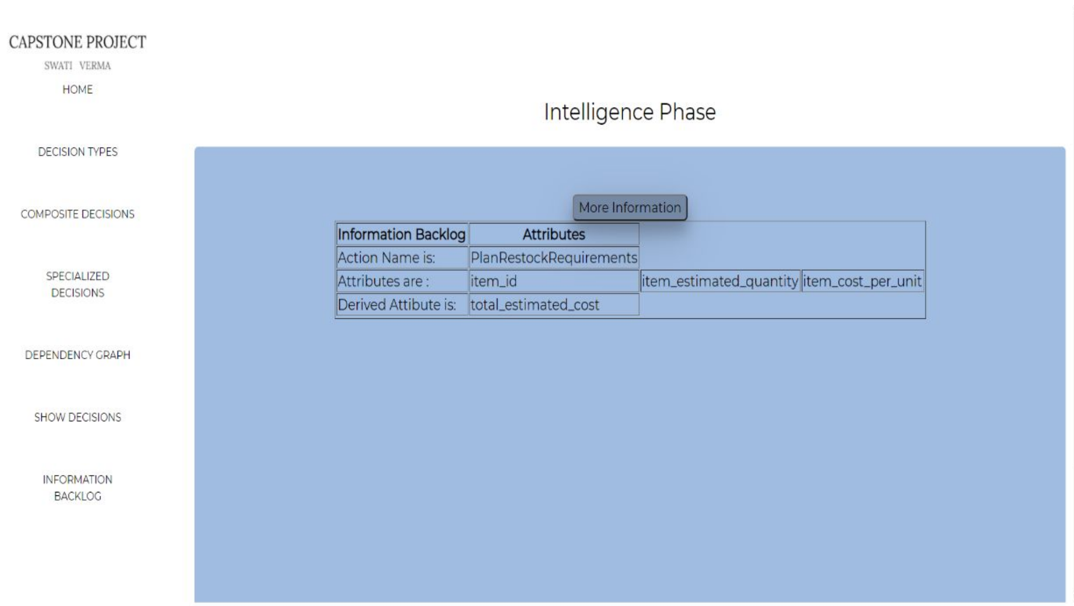

Fig 5.2 Intelligence Phase Page

In the below example initially the machine wise load is shown along with its attributes.

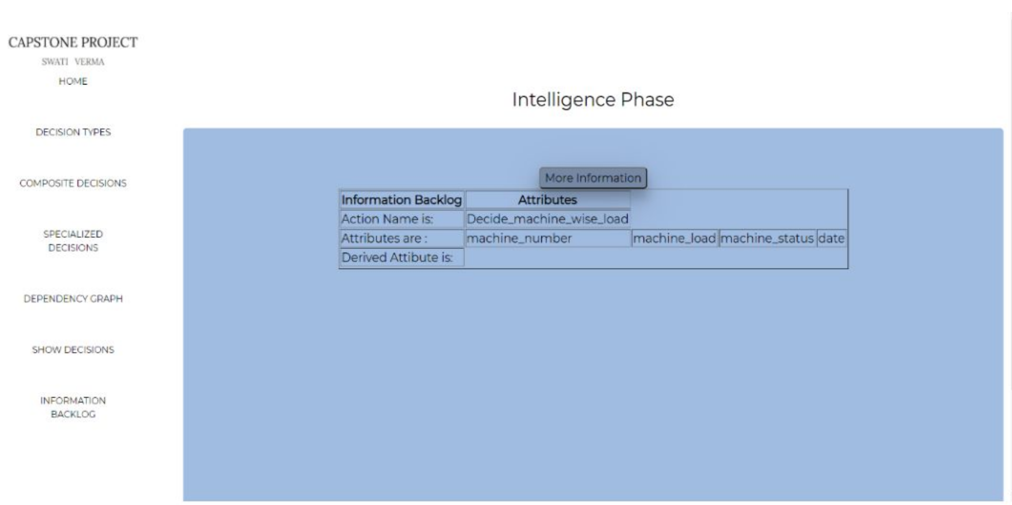

Fig 5.3 More Information Page

After clicking on more information, the other action accessing its attribute will be added in the information backlog.

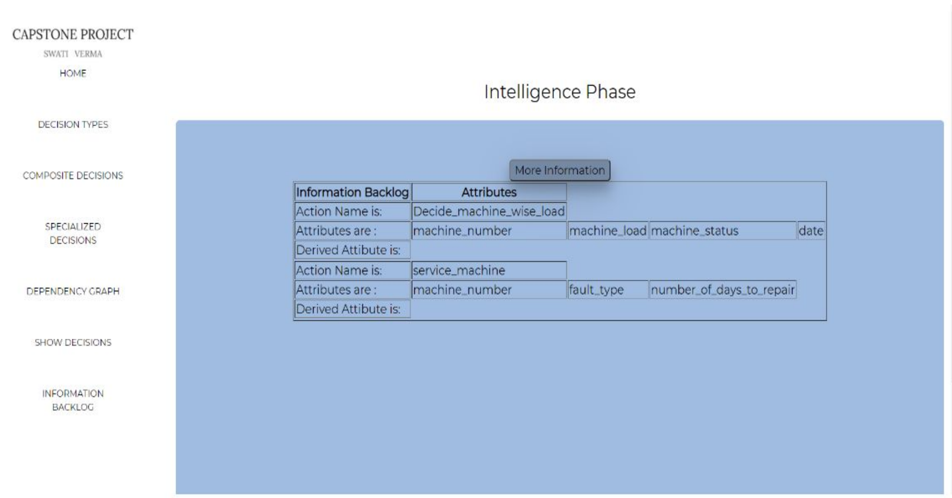

Fig 5.4 More Information Page

Choice Phase:

When the user selects the choice radio button then a new page is shown for the choice phase.

Initially all the alternatives are taken as input from the user. The alternative corresponding to the decision whether to accept the order or not is taken below.

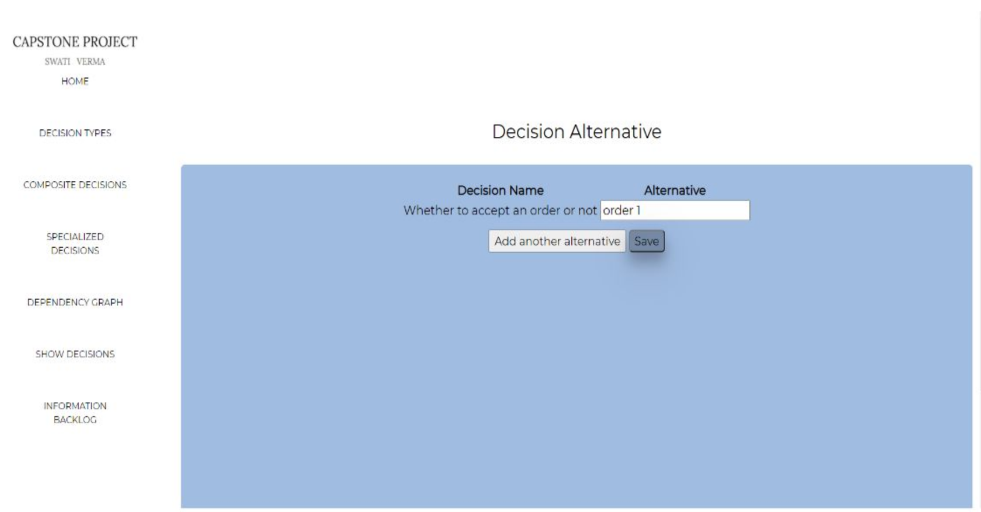

Fig 5.5 Decision Alternative Page

After the alternatives are taken, all the outcomes of one level are shown and their semantic priorities are taken as input which sums up to 1. If the sum is not equal to unity then the error message is also shown to enter the correct values.

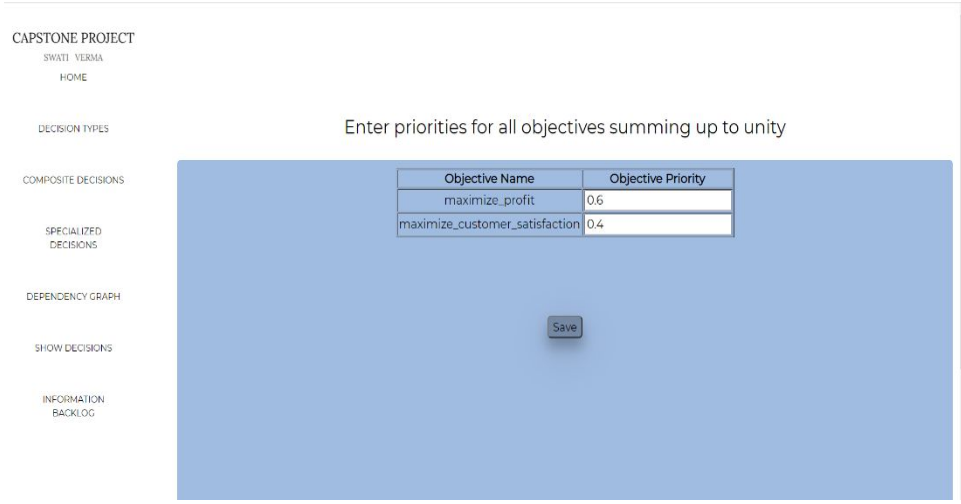

Fig 5.6 Get Objectives Priorities Page

After the save button is clicked the information size is estimated by taking a higher priority outcome first. Three options are given.

- Is small: If the size of information is too small then the next higher priority outcome is selected and the information size is estimated again.
- Is big: If the information size is too large then the set of all alternatives are given to choose a subset of alternatives which has to be considered for information estimation. And again the information size is calculated using this subset of alternatives.
- Is enough: If the user says the information size is enough for the sprint then all the computable attributes and alternatives are shown in the page.

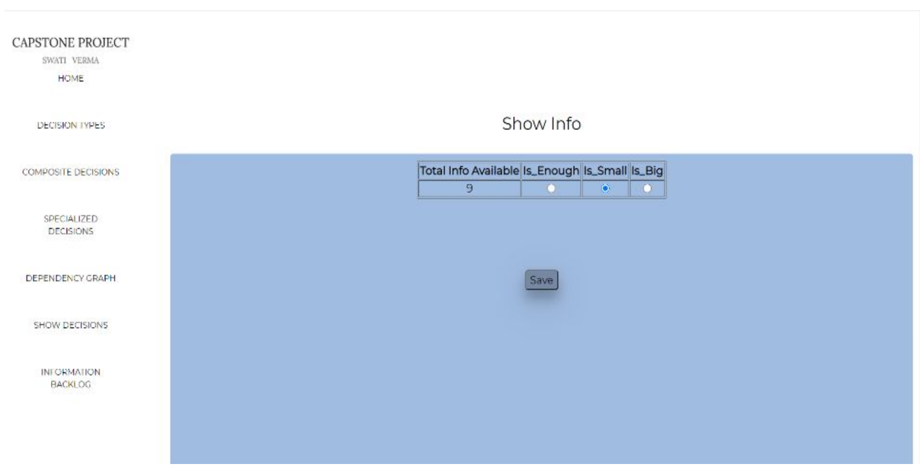

Fig 5.7 Show Information Page

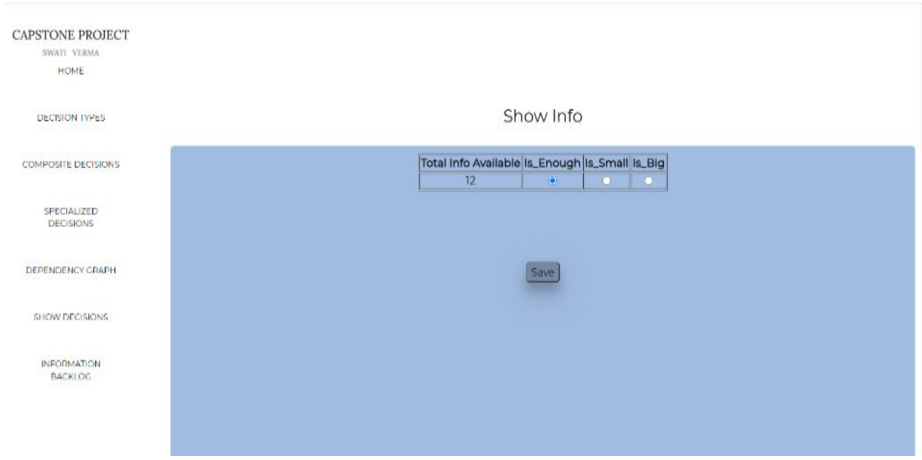

Fig 5.8 Show Information Page

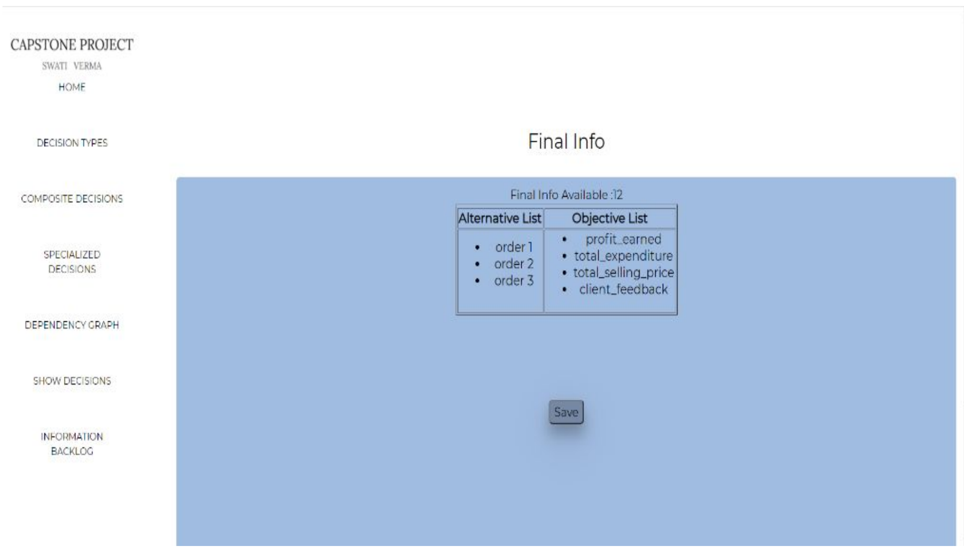

Fig 5.9 Show Information Page

6. Description of Tables

Different tables used in our tool are as follows:

1. **Decisions** Table: This table contains all the decisions and their ids.
2. **Actions**: This table contains action id, decision id and action name corresponding to a decision.
3. **Uncertainties**: This table contains uncertainty id and uncertainty name for all the decisions.
4. **Objectives**: This table contains objective id and objective names for all the decisions.
5. **Action_attr**: This table action id, attribute names, attribute data types and attribute types(base/derived) of all the actions.
6. **Action_attr_dep**: This table contains the dependency between base and derived attributes of actions.
7. **Objective_attr**: This table objective id, attribute names, attribute data types and isComputable attribute of all the objectives.
8. **Uncertainty_attr**: This table contains all the attribute names and their data types for all the uncertainties.
9. **Free_decisions**: This table contains the name of all the free decisions, their semantic priorities and is_visited which is true if the free decision is already selected from the backlog of decisions else it is false initially.
10. **Composite_decision:** This table contains all the decisions which belong to the composite backlog. The columns in this table are decision id, decision name, decision_syn_priority, semantic priority, is_parent which is true when the decision is parent or any other decision, is_root which is true when the decision is root of the composite hierarchy, parent name which contains the name of the parent decision and is_visited which is true when the decision is selected from the backlog.
11. **Specialized_decision:** This table is the same as a composite decision table and stores information about specialized backlog decisions.
12. **Final_dep:** This table contains all the dependencies of the dependency decision graph.
13. **Backlog**: This table contains the decision names, their Ssyn priority, Csyn priority, semantic priority and type of the decisions (free/composite/specialized) which are currently present in the free backlog. At the end when all the decisions are selected and removed from the backlog, this table becomes empty.
14. **Current_tree_display**: This table contains all the information about the current tree hierarchy which is displayed on the screen.
15. **Final_decision_order:** This table stores the final order in which the decisions are selected from the backlog. In the information backlog part the decisions from this table are populated on the screen.
16. **Show_info:** This table contains the action id, action name, its base and derived attributes which are shown during the intelligence phase.

17. **Decision_objective_map:** This table stores the mapping between outcomes and decisions. It also contains the count of computable attributes needed for evaluation of the outcomes, the priorities of the outcomes and is_visited which is true when the objective is already considered for evaluation.
18. **Decision_alternative:** This table stored the alternative taken for the evaluation of the outcomes during the choice phase.

7. Conclusion

In this project three tools are implemented. DIEM tool for capturing the components of influence diagram, Decision backlog tool for selecting a decision from backlog of decisions and the Information backlog tool in which the user can see the current enterprise situation using the past action records and can choose the best alternative in the choice phase.

7.1 Comparison with Kimball's approach of Data warehouse design

Similar to Kimball's iterative-incremental approach of building a data warehouse, we have also implemented the tool for selecting decisions from a backlog of decisions using the same approach. Each of the decisions has been considered as an increment and the actions, uncertainties, and objectives corresponding to that decision are taken as iterations within that increment. The tool provides decisions incrementally one at a time. Delivering the decisions one at a time helps in reducing the delivery time. As in the incremental model, all the key requirements are gathered first, in this project also we have identified all the decisions and their corresponding actions, uncertainties and objectives and represented them in the form of influence diagrams. We have captured this information using the DIEM tool.

Unlike Kimball's approach which does not have any information about how the data marts should be prioritized and developed, this project provides a way to overcome this problem by assigning semantic and syntactic priorities to the decisions. These priorities help in deciding which decisions should be taken next.

7.2 Applying Agile Backlog Approach to Sigma

One of the values of the Agile manifesto is **Individuals and Interactions Over Processes and Tools** which focuses on giving importance to individuals rather than tools because it is the people who understand the tool and can respond to the business requirements whenever needed. In this project, we have involved the family members of Sigma enterprise at each and every stage so that any change required can be incorporated easily and continuously. Rather than involving customers before and after the project, agile methodology involves **Customer Collaboration** during all the phases of the project. We have followed the same approach of customer collaboration by including stakeholders throughout the project by taking priorities while selecting decisions from the decision backlog and taking priorities of outcomes and asking if the information size is big, small or enough during the choice phase. During the intelligence phase also, the user is asked if they want to get more information about the decision.The **Responding to Change over following a plan**[14] has also been followed in this project by continuously incorporating changes asked by the stakeholders.

Agile principles like customer satisfaction through frequent delivery of the software, face to face interaction with stakeholders throughout the project, maintaining simplicity by delivering working software continuously and dividing the decisions into smaller increments are all followed in this project. Interaction through the stakeholders is given importance in this project so that the stakeholder can get information timely and can ask for the changes in the early stage of the data warehouse development. Prioritizing the decisions for reducing lead time has also been considered in this project

7.3 Challenges Faced

The major challenge faced initially was to understand the entire workflow of the Sigma enterprise, how they work, what are the major decisions taken by people involved in the decision-making process of the company. After the decisions are identified, the next challenge was to collect all the relevant information required by them in making these decisions. When taking the decision, the person tends to consider only a few factors at that time, but when we all sat down to identify them it was a time-consuming task to identify actions, uncertainties and objectives corresponding to each decision. After that, deciding the dependencies between different decisions, and categorizing them into free, composite and specialized backlogs was also time taking. After getting all the information the tool has been implemented. Understanding and implementing an algorithm which takes input from the user at run time and shows decisions and other information immediately was also challenging.

7.4 Future Work

As Agile methodology is taken in this project, in future also continuous interaction will be done with the members of Sigma Enterprise to understand the difficulties they are facing and if any improvement or addition of new features is asked by them then they would be incorporated. The other future work can be to deploy the tool as a software on their systems and not just run it on a local host. This would make the tool even more friendly and easy to use by the members of Sigma Enterprise.

List of References

[1] Prakash, Naveen & Prakash, Deepika. (2019). Handling the Information Backlog for Data Warehouse Development. 10.1007/978-3-030-27615-7_28.

[2] http://agiledata.org/essays/dataWarehousingBestPractices.html

[3] https://magnitude.com/blog/iterative-incremental-warehouse-development/

[4] https://tdan.com/data-warehouse-design-inmon-versus-kimball/20300#

[5]https://www.computerweekly.com/tip/Inmon-or-Kimball-Which-approach-is-suitable-for-your_data-warehouse

[6] https://www.educba.com/kimball-vs-inmon/

[7]https://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/enterprise-data-warehouse-bus-architecture/

[8] https://www.astera.com/type/blog/data-warehouse-concepts/

[9] http://thedataworld.blogspot.com/2014/02/enterprise-data-warehouse-bus.html

[10] https://unitfly.com/blog/decision-making-process/

[11] http://www.forestdss.org/wiki/index.php?title=Simon%27s_decision_making_model

[12] http://mytypings.com/herbert-simon-model-on-decision-making/

[13] Data Warehouse lecture slides by Dr. Naveen Prakash.

[14] https://www.smartsheet.com/comprehensive-guide-values-principles-agile-manifesto

Appendix

How to install and run the tool

1. Install python version 3.7 or more.
2. Install the python library Flask.
3. Install XAMPP and copy the project folder inside the htdocs folder of the XAMPP folder.
4. Open XAMPP control panel and start MySQL server.
5. Open phpMyAdmin and import the database. The mysql file is attached inside the MT19073_SwatiVerma_Capstone folder.
6. Open a new terminal and go to the directory where you project is located using change directory command.
7. You need to make some changes in the final.py file. You have to change the username and password of your MySQL server.
8. In the command prompt type

   > FLASK_APP= final.py flask run

9. After the final.py starts running go to the browser and type the name of the local server in which your app is running, for example http://127.0.0.1:5000/ and the website will start running in your application in the browser.

Workflow of the tool

1. **DIEM Tool**
   - First store decisions.
   - Store actions, action attributes and their types.
   - Store uncertainties, uncertainty attributes and their types.
   - Store objectives, objective attributes and their types.
2. **Decision Backlog Tool**
   - Select and save the type of backlog (free, composite, specialized) for each decision.
   - Store composite and specialized dependencies.
   - Store decision dependency graph.
   - In the show decisions page, enter the priorities and click on the show next decision.
3. **Information Backlog tool**
   - Select intelligence or choice for the decision for which you want to see the information.
   - In the choice phase, enter the alternatives and priorities for the outcomes to calculate the information size. In the intelligence phase, click on more information to get more actions and their attributes.