A 5 LAYER INFORMATION EXPRESSIVITY MODEL

APPLIED TO

SEMANTIC HETEROGENEITY IN

A DECENTRALISED ORGANISATION

M Dixon* System Development Group, Faculty of Information Systems,
Beckett Park, Leeds Metropolitan University LEEDS LS6 3QS

and

John Kalmus and Keith Jeffery Systems Engineering Division,
Rutherford Appleton Laboratory, Chilton, DIDCOT OX11 0QX

* also Visitor to System Engineering Division, RAL.

ABSTRACT

This paper discusses some issues associated with systems evolution in a decentralised organisation. The divisions of the particular organisation are functionally very similar but territorially decentralised and managerially autonomous. The paper considers the real case of an engineering fault logging system which has different versions in use by the different divisions of the organisation. It reviews the limitations of the standard architechtural approachs in the light of work on semantic heterogeneity. A recently proposed five layer model of information expressivity provides the theoretical framework for classifying the system variants. This approach allows us to focus on the layer requiring essential changes when seeking to interoperate these systems. We have chosen two examples from our case study; one example requires changes beginning at the semantic (business rule) layer and the other requires changes beginning at the schema (attribute representation) layer. The examples show the success of the five layer model in disentangling the semantics involved in this type of version change system interoperation.

1 INTRODUCTION

1.1 Introduction

Many organisations adopt computer systems to meet the operational requirements of particular autonomous divisions. In order to improve the organisation's effectiveness and efficiency there is a need for the operational data to be managed as an information resource for tactical planning purposes. This requires that the data in the different systems can be analysed together. The technical problems associated with combining information systems are well understood and appear to be tractable. However the semantic issues that arise appear to raise deep problems which require further investigation. At first sight it seems reasonable to assume that organisations which share the same world view can interoperate semantically. In this paper we consider a very simple example of a decentralised organsation for which the world views of the different divisions are very close. The divisions of the organisation have more or less the same business function and the information systems they use have been based on the same product from the same supplier; the systems have been introduced on a rolling basis across the organisation according to the plans of each divisional management by the same central service section. Yet even in this nearly ideal circumstance issues of semantic heterogeneity mean that analysing the data across the organisation for tactical planning purposes has to overcome significant obstacles. In order to probe the semantic issues we apply an information expressivity model to isolate where within the various levels of the information system the semantic differences arise.

1.2 Architectures

There have been two main approaches to designing interoperating heterogeneous databases: global schemas and multidatabase languages [BriHurPak92]. The global schema approach may be thought of as a direct outgrowth of distributed databases in which there is a global schema level generated from the local schemas. Sheth and Larson have defined a five level schema architecture which is now used by most workers as a reference schema architecture. [SheLar90, Sal&al94]. The alternative multidatabase language approach places most of the responsibility for integration on the users but provides them with support functions to ease their task [LitAbd87].

Coulomb and Orlowska noted that a key assumption [SheLar90] in much of the work on database interoperability was that local systems should retain their autonomy. However they showed by example that design autonomy was not generally sustainable because of semantic heterogeneity [CouOrl93]; interoperation is obtained by an extension to the schema. They argue that this necessary violation of design autonomy because of semantic heterogeneity calls into question the utility of research on the general architectural approach except for the type of system they classify as public reporting; these latter systems have published schema and are subject to query only access. Many specific examples of interoperation exist but this arises from the fact semantic heterogeneity can be resolved for interoperation where organisations share the same world view.

Jeffery et al [Jef&al94] have been studying a multi media information processing system and introduced knowledge based assists to reconcile schema and to integrate the queries returned from different databases. Part of the theoretical framework to tackle the database semantics was the introduction of an architecture based on a five layer model of information expressivity. Their five layer model is an extension of the traditional three layer data modelling approach of having conceptual, logical, and physical layers. They explain the need for the extra layers by their view that the traditional approach is weak in two areas 1) the mapping of the business requirements onto some kind of entity relationship diagram and dataflow diagrams fails to capture the full richness of the semantics of the end user and domain requirements. 2) interaction of processes with data; these involve constraints and triggers. Non functional requirements at the conceptual level can also be expressed as constraints. It is important to distinguish this 5 layer information expressivity model from the 5 layer schema model of Sheth and Larson[SheLar90]; the latter is an extension to heterogeneous databases of the ANSI/SPARC system configuration architecture [ANSI75].

Methods of automatically resolving semantic heterogeneity in multidatabase systems have been suggested. The Summary Schema Model, SSM, aims at linguistic support to automatically identify similar entities. It aims at letting end users employ their own terminology and perhaps formulate an imprecise query. The SSM uses a hierarchically global data structure combining dictionary and thesaurus. The semantic distance metric is introduced as a quantitative measure of semantic difference; it is based on a weighted count of the links in the path between two terms.[BriHurPak94]

1.3 Summary of our Decentralised Organisations and Systems

The case study used here is based upon an airport which had four Terminals handling passengers and aircraft. Within each Terminal there is a need to carry out repairs on faulty equipment and to schedule preventitive maintenence. For practical purposes each of the Terminals at the airport had its own management structure, cost centres, and purchasing decisions. Collaboration and cooperation between the Terminals had to be negotiated. There was a central services division which coordinated development of information systems, was responsible for their installation and systems work; the Terminals were responsible for phasing of procurement and the operational use of the system. At a practical level this meant that some Terminals were running a different version of the information system to others. The software was run on compatible although not identical minicomputers; for the purpose of this paper we will ignore upversioning of the dbms which was a proprietary product of one software house.

1.4 Purpose of Study

The purpose of this study is to investigate whether using the 5 layer information expressivity model can benefit the resolution of semantic heterogeneity in a decentralised system described by us elsewhere[DixKal94]. The business functions were located in functionally similar but territorially distinct divisions of an organisation; the divisions were relatively autonomous but use similar information systems. We are particularly concerned with the suggestion from Coulomb and Orlowska that interoperation of databases involves extension to the schema semantics as a consequence of querying. We shall look at whether a different query might require a different extension. Coulomb and Orlowska's original formulation was concerned with key identification within different systems. This work will consider another type of extension which relates to the querying itself changing the business rules. We will focus on system evolution since version variation has an important impact and is often introduced in order to improve organisational efficiency in enterprises.

2 5 LAYER MODEL OF INFORMATION EXPRESSIVITY

2.1 Introduction

In this section we indicate the way that we partition our description of the different versions of the information systems following the 5 layer model of information expressivity. [Jef&al94]. It is important to note that we are not attempting to describe the original information systems in terms of the 5 layer model; we are trying to focus only on those features which are of interest to the application of queries relevant to interoperation of version changed systems.

2.2 Semantic Layer

In the semantic layer we define the business rules that we are considering. The terms will be defined in the way that the business sees them using a structured English.

2.3 Conceptual Layer

The conceptual modelling process is one of expressing the business in terms of one specific data model. Different data modelling approachs use different tool boxes and hence yield different descriptions. We propose to use SSADM in constructing our data model since this methodology is often a central government requirement on its contracts in the UK. The data model involves an Entity Relationship model, Entity Life History models, and Data Flow models in which the processes are described in structured mini specifications.

2.4 Intensional Layer

The intensional layer of the information system expresses the constraints that apply to the data values of the attributes. Within databases the constraints can be represented by triggers which are invoked by the database management system; typically these are embedded in application code attached to forms which are used for data manipulation. These constraints may reflect the domain of the attribute or depend on the values of other attributes. The satisfaction of a constraint may lead to the resetting of attribute values. Access to certain data values may be restricted so that only designated users could modify those data values.

2.5 Logical Layer

The logical layer expresses the conceptual model in terms of the underlying data model of the database management system. This level is expressed in terms of the schema of the database. For the system we are considering this was the relational data model; we therefore assume the tables can be mapped to a union compatible form across the version change and a query applied using the UNION operator. It is important to recognize that the RM is different from the (E)ERM [Sal&al91] although some authors consider the latter to be a thin skin over the former.

2.6 Physical Layer

The physical layer describes the way that the model is implemented physically on the platform; we do not intend to discuss this layer further here.

3 CASE STUDY

3.1 The case description

In the Appendices we give the entity, attributes, and attribute definitions for the old (Appendix 1) and new (Appendix 2) versions of the fault logging system as implemented. Further details of the case study may be found elsewhere [DixKal94, Framp86]. We have selected two examples from the case for study in order to show how the 5 layer information expressivity model allows us to focus at the appropriate information level. One example has no change to the schema but there is a substantial change to the business rules. For the other example the schema are radically different in the different versions but represent the same business rules.

3.2 Reasons for wanting interoperation

Here it is important to recall that the systems were developed and installed to meet the operational requirements of the maintenance crew; the lifetime of most faults is measured in hours rather than weeks. Analysing the information across different versions is seen as an additional requirement used to provide a) historical continuity within a Terminal b) patterns of reliability across the airport c) tracking of plant transfers between Terminals d) comparison of performance on maintenance effectiveness

4 EXAMPLE 1 OF SPECIFICATION IN 5 LAYER MODEL

4.1 Introduction

Some of the issues associated with effects of changing the versions of a database have been discussed in our recent paper which used the engineering maintenance database as a demonstrator system.[DixKal94]. That paper considered the effects of two seperate changes to the semantics associated with the time an item of plant was out of service due to plant breakdown; there was an attribute on each fault log record named Repair Time. It is the purpose of this example to fit the semantics within the 5 layer model of information expressivity [Jef&al94]. We will consider the Repair Time of a given item of plant in three database representations:

S1) The old fault logging database for which the Repair Time was calculated from the fault reported and fault clearance times

S2) The new fault logging database for which the Repair Time took account of the time each day when that item of plant would not be required because of operational practice at a Terminal such as a ban on night flying

S3) The item of plant is moved to another Terminal which has a different pattern of operational practice and so has a different window of required availability. A similar effect could arise if a Terminal changed its operational practice.

These different representations do not cause an operational problem to the maintenance crews for whom the fault logging systems were installed to aid their engineering activites in clearing faults. The difficulties arise when the data is used for tactical planning purposes eg assessing the reliability of equipment or the efficiency of maintenence crews. In these cases the semantic differences could radically affect the reliability of the planning conclusions. Note that in the case of representation S3 that the semantic heterogeneity can arise without changing the local database system but only changing the data within the system.

4.2 Semantic Layer

4.2.1 The business models

S1) Repair Time is the time interval between a fault being reported to the maintenance crew and when the plant item can be used again.

Repair Time calculated as (Fault Clearence Date/Time - Fault Report Date/Time) Fault Report Date/Time is time stamped with system time when fault on plant is first reported. Fault Clearence Date/Time is data reported by maintenance crew.

Comments: All faults reported to the maintenence crew are recorded quickly on the database so Fault Report Date/Time is a fair starting point. Simple correspondance between data in fault log record and the semantics.

Anomolies: Computer system down when fault report made. Fault clearence extends over BST/GMT change.

S2) Repair Time is the amount of time between a fault being reported to the maintenance crew and when the plant item can be used again after the time for which it is not required to be available is excluded.

Repair Time calculated as

(Fault Clearence Date/Time - Fault Report Date/Time - Plant Unrequired time)

Fault Report Date/Time is time stamped with system time when fault on plant is first reported. Fault Clearence Date/Time is data reported by maintenance crew.

Plant Unrequired time is

SUM over each day (Daily Plant Unrequired time)

where

Daily Plant Unrequired time

= (EndNotInNeed - BeginNotNeed)

and

(BeginNotNeed >= Fault Report Time

and

EndNotInNeed <= Fault Clearance Time

and

EndNotInNeed corrected for day change.)

Comments: All faults reported to the maintenence crew are recorded quickly on the database so Fault Report Date/Time is a fair starting point. Date/Time is no longer atomic; seperate arithmetic is applied to the Time field so two different fields are required. Anomolies: Computer system down when fault report made. Fault clearence extends over BST/GMT change.

S3) is S2) with different data.

4.2.2 ‘Schema’ Extension of Business Models

An important feature of the Coulomb and Orlowska approach to interoperation is the recognition that this leads to a schema extension. With the present example of semantic heterogeneity we define schema extensions of the Business Models to encompass S1), S2) and S3).

1) S1 can be regarded as a limiting case of S2 for which the Daily Plant Unrequired Time is set to zero.

2) S2 can be regarded as a specialisation of S3; for S3 the different values Daily Plant Unrequired Time were used.

Now interoperation of the databases can be obtained with consistent semantics by one of several strategies

a) Re interpret all Repair Time values as though Daily Plant Required Time were zero. This is a numerically exact solution to providing semantic consistency. However it is an undesirable solution for a business indicator since it might lead to unnecessary effort to improve the indicator while not benefitting the business.

b) Provide estimates for the unavailable BeginNotNeed and EndNotInNeed data values. Even if the data is not precisely accurate it may provide a better business indicator for tactical planning.

c) Provide a time history for the unavailable BeginNotNeed and EndNotInNeed data values.

It is important to note that the semantics associated with the extension of S1 for S2 is determined by the purpose of the interoperation; a different purpose, say allowing for the delay in waiting for replacement parts from a supplier, would lead to a different extension to S1.

4.2.3 Conclusions

In this example we mapped the Repair Time of the fault logs onto the 5 layer information expressivity model. We discovered the following:

1) In order to interoperate databases it must be possible to construct a schema extension of the seperate business models that allows a consistent semantic interpretation to be defined. In our case we needed formulate an interpretation of Repair Time which could be used in the particular querying context for each of the three versions

2) The population of the extended model requires business rules which need to be decided at the semantic level. The rules need to be determined by their use as business indicators rather than just providing a numerically accurate way of generating consistent semantics. In our case using guessed data, based say on current practice, might be a much better guide than the historic data values for the calculated field in the database

3) There is not a single extension which can be identified rather the nature of the extension to the schema comes from the semantics required for interoperation. A corollary of this is that for this type of semantic extension, involving essentially a transformation, the use of thesaurus type techniques would not be expected to be fruitful.

4.3 Other Layers

Changes to the business rules must be propagated down to the lower layers of information expressivity. Details of the conceptual, intensional, and logical layers are given in Appendix 3.

5 EXAMPLE 2 OF SPECIFICATION IN 5 LAYER MODEL

5.1 Introduction

Maintenance crew often have to replace components within the plant in order to repair a fault. Work carried out during preventative maintenance involves the pre issue of a known set of parts which are identified in the plant's bill of materials. However in processing a breakdown fault the maintenance crew need to select the parts to be replaced; in reporting their choice they may use a part description or common name rather than the full part code. The difference in information system versions arises because for the older version of the system the parts used were collected as a simple text field whereas for the revised system the parts were individually identified.

S1) The parts and the quantity of those parts which are replaced in the repair of a fault are recorded in a Parts Used description field. The maintenance crew may use a common part name as a description to identify the relevant part. The information is recorded as a free text field seperated by terminators for each fault log.

Parts Used = { part term}
part term = [quantity + [x]]+ (part number|part description|common name) +[x] [quantity] + [terminator]
terminator = ([ ]|.|,|) x = multiply by

S2) The parts used in the repair of a fault are identified in a Parts Used field by a standardised part number selected from the components list of the plant item. The quantity of the part used is collected as a seperate field. Not all repairs are carried out using the designated components list; sometimes an alternative may be just as suitable and this may be identified by the crew using a part description or a common name.

Parts Used = (part number|part description|common name)

Part Quantity = (quantity)

5.2 Semantic Layer

The business models for both systems are intended to be more or less identical as far as querying is concerned; the differences relate to the format of data entry.

5.3 Conceptual Layer

Please see appendices

5.4 Intensional Layer

5.4.1 Constraints on S1

No specific constraints exist in S1 since the Parts Used field here is a free text field associated with the i'th fault log, FL(i). However there is the implicit assumption that

Parts Used (i) |-> {Parts Used(i,j), Part Quantity(i,j)}

where Part Used (i,j) is the j'th Part Used to repair FL(i)

forall FL(i) ((Parts Used(i,j) ¬=null)
=>((Part Quantity(i,j) ¬=null) ^ (Part Quantity(i,j) >0))

5.4.2 Constraints on S2

Part Used (i,j) is the j'th Part Used to repair the i'th Fault Log, FL(i)
forall FL(i)
((Parts Used(i,j) ¬=null)=>(Part Quantity(i,j) ¬=null) ^ (Part Quantity(i,j) >0))

5.4.3 Additional Constraints for Interoperation

In order to be able to compare exactly the data relating to parts in S1 with the data in S2 it must be possible to uniquely associate the parts identified in S1 with a corresponding part in S2. In practice this was not feasible because of the use of common names in both S1 and S2 so the constraint is relaxed to the values being drawn from domains that can be mapped onto one another. We will discuss the issues associated with non unique mapping subsequently.

Domain ({S1_Parts Used}) |-> Domain ({S2_Parts Used})

5.4.4 Non Unique Mapping

In this section we assume that the user has correctly captured the intended part name. However if the name is ambiguous it cannot be used to exactly identify a part held in stores or by a supplier. Because we are dealing with a system which has been in use for many years we do not have the opportunity to go and retrospectively observe the replacement part since it itself could well have been replaced in a subsequent maintenence visit. Within the relational model the issue is equivalent to the disambiguation of foreign keys and is not peculiar to interoperation. Several different approaches are possible but they depend upon what is the purpose of the queries that are to be posed.

a) ignore the ambiguity: it may not be significant since for practical purposes the parts are interchangeable eg batteries from different manufacturers. This approach is best suited to the case where failure is not specific to a particular part type.

b) use a probabilty factor: use the purchasing records to identify the quantity of different parts obtained for the relevant time interval. This approach would not be appropriate if the purpose of the search was to look for differentially defective sources of parts.

c) examine associated text in the fault log: to see whether additional information is recorded. This may be combined with anecdotal information from engineering crew.

5.5 Logical Layer

5.5.1 Database System Models

The difference between the two versions is recorded at the logical level because the older version is not genuinely a relational database and the underlying data model was a flat file.ls .

5.5.2 Creating a logical schema

Here we need to create a new temporary table for the length of the querying session and then populate it from the older version of the database. The population would need to be carried out by application code equivalent to a trigger defined on S1_Parts Used in 5.4.2 above at the intensional level.

create table temporary S1_partsused

(S1_Part_Number datatype(S2_Part_Number not null),

S1_Fault_Number datatype(S2_Fault_Number not null),

S1_Part_Quantity datatype(S2_Part_Quantity not null))

5.6 Physical Layer

Not applied.

6 DISCUSSION

We have discussed the difficulties facing decentralised organisations in achieving cooperation between their information resources. The consequences of semantic heterogeneity led Coulomb and Orlowska to propose that it was necessary to define a schema extension to achieve interoperation. This present study applied the 5 layer information expressivity model of Jeffery et al to classify the nature of that extension for non update querying in the case of system evolution.

One example from the case study hightlighted the change in business rules while the schema remained identical; this work emphasised the importance of going back to the business rules when considering interoperation. The example was also used to show that the queryer may redefine the business rules and apply that redefinition to both versions.

The other example from the case study showed that even where the schema were radically different they could be representing the same business rules.

We conclude that a benefit of the 5 layer model is that it seperates out a change in the business rules for systems on interoperating from changes at the database implementation level. This allows us to distinguish between what can be solved using computing technology and what requires further business input for interoperation.

We are led to a wide rather than a narrow definition in interpreting Coulomb and Orlowska's term schema extension.

From the two examples we conclude that the system modelling process is considerably enriched by using the 5 layer information expressivity model and are encouraged to explore further the benefits of using it.

REFERENCES

ANSI75 ANSI/XS/SPARC: Interim Report from Study Group on Database Management Systems Bull ACM SIGMOD 7,2 1975

BriHurPak92 M W Bright, A R Hurson, and S H Pakzad A taxonomy and Current Issues in Multidatabase Systems Computer, vol 25, p50, 1992.

BriHurPak94 Automated Resolution of Semantic Heterogeneity in Multidatabases M W Bright, A R Hurson, and S Pakzad ACM Transactions on Databases 19, 212, 1994

CouOrl93 R M Coulomb and Maria Orlowska, Interoperability in Information Systems,Technical Report 263, Department of Computer Science, University of Queensland, 1993.

DixKal94 Semantic Heterogeneity in Interoperating Databases: An Investigation based upon Engineering Maintenence M Dixon and J Kalmus, Proceedings of 6th ERCIM Database Research Group Workshop on Deductive and Interoperable Databases, (Barcelona), 1994

Framp86 A Frampton Planned Maintenance System, Report BAAPMS.Z1, DM England & Partners, 1986

Jef&al94 K G Jeffery, L Hutchinson, J Kalmus, M Wilson, W Behrendt, and C Macnee, A model for heterogeneous distributed database systems, BNCOD 12, Springer Verlag Lecture Notes in Computer Science 828, p221, 1994

LitAbd87 An Overview of the Multidatabase Manipulation Language MDSL, W Litwin and A Abdellatif Proc IEEE Vol 75(5), 1987, p62

Sal&al91 On Canonical Models for Federated DBs, F Saltor, M Castellanos and M Garcia-Solaco Sigmod Record Vol 20(4), 1991, p 44

Sal&al94 On Architectures for Federated DB Systems F.Saltor, B Campderrich, M Garica-Solaco Proceedings of 6th ERCIM Database Research Group Workshop on Deductive and Interoperable Databases, (Barcelona), 1994

SheLar90 A P Sheth and J A Larson, Federated Database Systems for managing distributed, heterorogeous and autonomous databases.ACM Computing Surveys 22(3), p183, 1990.

ACRONYMS

BST/GMT British Summer Time/Grenwich Mean Time

(E)ERM (Extended) Entity Relationship Model

RM Relational Model

SSADM Structured System Analysis and Design Methodology

SSM Summary Schema Model

APPENDIX 1

Definition of Old Fault Log Entity

Attribute	**Definition**
Fault Number	System issued unique number
Report Time	Date/Time when fault logged; autostamped
Plant Name	Unique name for plant item
Plant Group	Category of Plant
Location	Place where plant is located in Terminal
Area	Grid value
Cost Code	Head of charge for work
Fault Title	Summary of Description of Fault
Fault Descrip	Full description of fault
Crew	Maintenance Crew responsible for work
Report by	Person issuing initial fault report
Clearence	Action and comments taken to clear fault
Future Work	Engineers view of further work needed on it
Complete Time	Date/Time engineers report fault cleared
Repair Time	Time out of service
PartsUsed	A list of parts used in repair

APPENDIX 2

Defintion of Plant Entity

Attribute	Definition
Plant Number	Unique Nato code for plant
PlantDescript	Working name for plant; unique
Plant Group	Type of plant
Location	Place where plant is in terminal
Area	Grid reference
Cost code	Head of charge for work
DateInstalled	When plant was installed at airport
Manufacturer	Name/address of plant manufacturer
BeginNotNeed	Begin time when plant would not be needed
EndNotInNeed	Time when plant needed again

Definition of New Fault Log Entity

Attribute	Definition
Fault Number	System issued unique number within year
Fault Year	Code for year; unique fault year / number
Report Time	Date/Time when fault logged; autostamped
Plant Numer	Unique NATO code; foreign key
Fault Title	Summary of Description of Fault
Fault Descrip	Full description of fault
Crew	Maintenance Crew responsible for work
Report by	Person issuing initial fault report
Clearence	Action and comments taken to clear fault
Future Work	Engineers view of further work needed on it
Complete Time	Date/Time engineers report fault cleared
Repair Time	Time out of service
PartsUsed	A list of parts used in repair

Definition of Parts in Store Entity

Attribute	Definition
PartNumber	Nato code; unique
PartDescrip	Common name for part; non unique
StoreLocation	Which stores part is held
StockLevel	Count of parts in stock
ReorderLevel	Count of parts when should reorder

Definition of Parts Used on Fault Entity

Attribute	Definition
PartNumber	Nato code; unique
Fault Number	Fault on which part was used; foreign key
Part Quantity	Count of this part used on this fault
WorkOrderNum	Identifies work order part used on; unique

APPENDIX 3 Repair Times Conceptual, Intensional and Logical Layers

A3.1 Conceptual Layer

See Appendix 4 for the reconciled entity relationship diagram. We do not report either data flow or entity life history diagrams because the query is not updating data. Only the reconciled diagram is given since the query has to be against a consistent conceptual representation. A representation is used that takes account of the generalisation necessary for more than one window of non availability for the plant.

A3.2 Intensional Layer

A3.2.1 Constraints on S1

We use ¬null here to mean the value is not null in contradisction to its use in the relational model where it returns false.

For the i ‘th fault FL (i) S1TV(i) = ((Report Time (i) = ¬null) ^(Completion Time (i) = ¬null) ^(Completion Time (i) > Report Time (i)) ^(Repair Time (i) = (Completion Time (i) - Report Time (i))). forall FL(i) (there exists (Repair Time) iff S1TV(i))

Privacy: Report Time(i) autofill to System Time; Repair Time(i) calculated ^ not editable

A3.2.2 Constraints on S2

For the i ‘th fault FL (i) S2TV1(i) = ((Report Time (i) = ¬null) ^(Completion Time (i) = ¬null) ^(Completion Time (i) > Report Time (i))). For plant item p = Plant Number (FL (i)) S2TV2(p) = ((BNN(p) = ¬null) ^(ENN(p) = ¬null). Here we have used an obvious shorthand of BNN and ENN for BeginNotNeed and EndNotInNeed respectively.

(intervalPerday = (24 + (ENN(p) - BNN(p))) ^S2TV2(p)^(BNN > ENN)) V (intervalPerday = ((ENN(p) - BNN(p)) ^S2TV2(p)^(BNN < ENN))

(interval1stday = BNN(p) - timepart(Report Time(i))) ^ S2TV2(p) ^(BNN(p) > timepart(Report Time(i)))) V (interval1stday = 0 ^ S2TV2(p) ^(BNN(p) <= timepart(Report Time(i)))

(intervallastday= timepart(Completion Time (i)) - ENN(p)) ^S2TV2(p) ^(timepart(Completion Time (i)) > ENN(p))) V (intervallastday= 0 ^S2TV2(p) ^(timepart(Completion Time (i)) <= ENN(p)))

(Repair Time (i) = (daypart(Completion Time (i) - daypart(Report Time (i)) X intervalPerDay) + interval1stday + intervallastday) ^S2TV2(p)) V (Repair Time (i) = (Completion Time (i) - Report Time (i))) ^¬S2TV2(p)

Privacy: Report Time autofilled Repair Time calculated ^ not editable

A3.3 Logical Layer

A3.3.1 Creating Logical Schema

Our example corresponds to the public reporting system paradigm because the extension is one sided applying only to the old version into which we incorporate information from the new version. In this section we will assume that the affected plant has been correctly identified by its plant number, later we will address the situation where this is not the case.

1) The first stage is equivalent to creating a view with the relevant columns

create view viewoldfault as select Faultno, Plantno, Report Date, Report Time, Completion Date, Completion Time, Repair Time from Fault

2) One extension to the schema of the old system can then be made by defining the window of non required availability to be the same as the current

create view oldversion as select Faultno, Plantno, Report Date, Report Time, Completion Date, Completion Time, Repair Time, BeginNotNeed, EndNotInNeed from Fault, PlantRegister where Fault.Plantno = PlantRegister.Plantno

The data values in Repair Time would require updating according to the formula in A3.2.2.

A3.4 Physical Layer

Not applied

APPENDIX 4

Figure 1: Entity Relationship Diagram of Conceptual Model for Repair Time allowing for the possibility of more than one out of service window per plant item

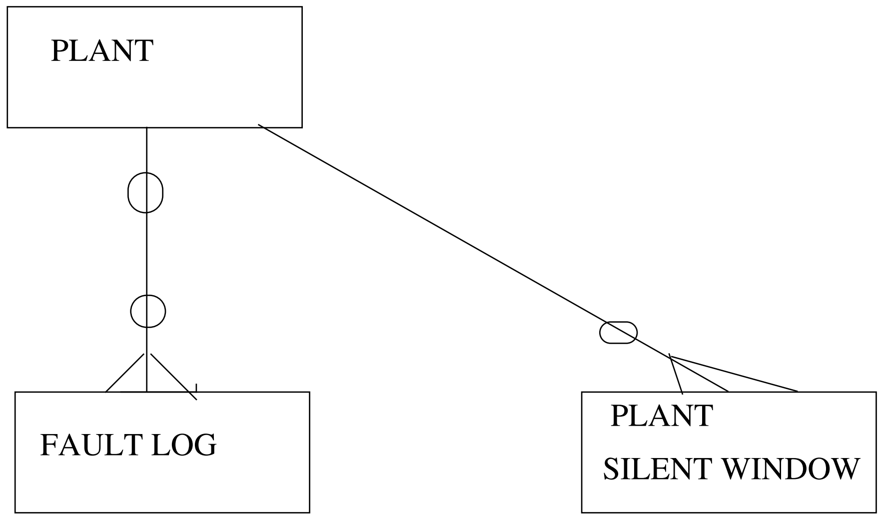

Figure 2: Entity Relationship Diagram of Conceptual Model for Parts Used in a fault repair.