
Information Resource Management

By Mike King (Mike King Consulting & Courses) In cooperation with Walter Webb (Eskom)
--

Invitation

It is the intention that the ideas expressed in this document will form the basis for a useful interaction between practitioners in the area, and it is hoped that by sharing each other's experience and ideas, a better insight into the 'STATE OF THE ART' may be built up.

With this objective, the reader is invited to respond to the points in the document, as a way of initiating some ongoing informal discussion.

The following headings might be relevant for each point (but add your own):

- Do you agree that this or some equivalent is the way to go (what equivalent)?
- Do you think that this is not the way to go (why)?
- Did you try this and experience difficulty (what difficulty)?
- Did you try this successfully (how did you succeed)?
- Do you recommend some other approach?
-

The following overall headings could guide the readers' general response (but add your own):

- Overall comments about the ideas in the document (especially in terms of the readers experience);
- What critical aspects have been omitted?
- What do you think are the potential 'show stoppers' to achieving and deriving value from an effective corporate resource of metadata?
- Do you agree that metadata has to play the central role in managing the information resource?

If you would rather have a presentation of the concepts (overhead projector etc.) and then have an informal discussion, this can be arranged. E-mail mcking@icon.co.za or telephone (011) 781 0444. This only applies to venues within reasonable distance from my office. In the case of more distant venues, an e-mail discussion could be initiated.

Introduction

Many attempts by organizations to implement Information Resource Management have failed. There have been numerous reasons, including:

- Tools (repository etc.) have often been inadequate for the job;
- The resource management process has sometimes become a bottleneck that slows down application development teams;

- The advantages of having a corporate information resource have not been accepted by all role players so that there has not been proper management support;
- The entity models have become points of view that have no correlation with the real systems being used;
- The repository information, being of low quality, adds no real value to any part of the business;
- ... And so on ... Not attempting to be complete here ***Please mention the reasons you experienced.***

This document is intended to single out ***those aspects of information resource management, which are critical to its successful implementation.*** No attempt is made to describe a complete framework. It omits the aspect of the need for committed top management support, as it is assumed that there will not be disagreement about this.

Also, the **document has focused mainly on the data.** It should be expanded to include all aspects (for instance: mission, objectives, functions, processes, CRUDS, modules, applications, ...). It is felt that the practical problems that block the achievement of the IRM objectives can all be addressed with this focus, and so if a success can be made of the data aspects, the other aspects would not be difficult to add.

Crucial Points from the Perspective of this Document.

The following headings correspond to the document layout.

Some Principles:

Is it the lack of attention to these principles that often leads to failure?

Services that add value to the business:

Are these necessary and sufficient?

Metadata:

Is the metadata really the crucial element of information resource management?

Content of the Repository:

What is the essential content of the repository to enable successful information management?

Cross-references between the actual files and databases, and the entity model:

How important is this for successful information resource management, and how much detail is necessary?

Quality state of the repository:

How does one measure this, and what actions does one take to improve it? How handle the baseline, and the planned baselines? (See paragraph on baseline)

Source of metadata in the repository:

Five sources are recognised. They are legacy systems, new systems being purchased or under development (three subcategories for this), and strategic planning exercises. Are there other sources?

Capture of metadata into the repository:

The framework for part of this is shown in the lifecycle diagrams. It is in this process that there is a risk of information resource management becoming a bottleneck, which slows down the systems development activities. The choice between an active and a passive repository is relevant to this. So is the effectiveness of the repository toolset.

IRD/IRDS:

These are repository standards. To what extent are they being used in practice?

CASE tool as repository:

Should the CASE tool being used for new systems development be used also as the repository to coordinate all projects, and all information in the organization? What are the main trends in this respect? What experience is available (failures as well as successes)?

Baseline:

This document implies that one of the main reasons for failure of information resource management is that there is not a clear definition of the meaning of the repository in respect of the time dimension. It is here assumed that it is unrealistic to cater only for a development, test and production repository! In reality there are continuous waves of change which are under development at the same time, but which are due to be implemented one **after** the other with months between them. All need to be quality assured in the context of the repository and yet should not become part of the baseline until they are implemented. In other words, the repository is defined to represent the business **as it is now** in respect of its mappings to databases and files, but to contain also future states of the enterprise in respect of unimplemented entities. Is this thinking right? What experience is available? What repository systems support this approach?

Some Principles

- Without the use of an effective repository of logical and physical models, the information resource management objectives are very unlikely to be met.
- Logical models, which are insufficiently representative of actual business data, will fall into disuse.
- The faithfulness of the logical models to the actual business data cannot be controlled unless the data model elements are cross-referenced to the actual details of the databases and kept up to date. Without this, there is no way to tell which parts of the data model represent the business as it actually is. I.e. it is important to distinguish between implemented and unimplemented parts of the data model. Also this enables the unexplained/unintegrated parts of the databases to be highlighted (tables, columns, files, fields which are not cross referenced to any data model element).
- Where duplicated data is kept, it should be consistent. This requires knowledge of its existence. The cross-reference between the logical data model and the existing databases documents this duplication when it exists.
- Logical models that are not explicitly mandated for use in applications development processes will fall into disuse because they will cease to represent the actual business data as it evolves.
- Data about the same concept (for instance data about Customers) should have the same structure and should be referred to by the same terms, no matter which part of the organization keeps it (at the least, synonyms should be documented in the repository).

Services that Add Value to the Business

The objectives of IRM include the supply of the following services:

Services That Use But Do Not Change The Repository Contents (meta-data)

- Support for the search for available existing business data (by end users)
- Support for the discovery of new opportunities to use existing business data (by end users)
- Support for the discovery of the precise detailed meaning of data elements within the business context (by end users, developers, data architects, process architects, application architects).
- Support for Impact Analysis
- Guidance to the End User in data quality control procedures (in addition to data model controlled integrity)
- Advice on whether a software package is likely to integrate with existing and proposed systems

Services that Use and Change the Repository Contents (meta-data)

- A cross reference between the Entity Model elements and Processes
- Cross-references between the Entity Model elements and the actual Tables/Files of the databases at all sites (Entities for which no cross reference exists, are regarded as unimplemented - tables/files for which no cross reference exists are regarded as outside the scope of the integrated corporate resource - unexplained/unintegrated).
- Cross-references between the attributes of the entities and the columns/fields of the tables/files. (Attributes for which no cross reference exists, are regarded as unimplemented - columns/fields for which no cross reference exists are regarded as outside the scope of the integrated corporate resource)
- Cross references between the Processes and the Applications
- Cross-references between the Applications and the Tables/Files.

Cross-references (mappings) between Entities and Tables/Files and between Applications and Tables/Files

Entity Name	(Entity Attribute = Table/File Column/Field)	Table or File Name
Partner	Partner No = Client No, Partner Name = Client Name, Contact Name = Client Contact.	Client (Oracle)
Partner	Partner No = CustomerN, Partner Name = CustName, Balance = Balance.	Customer (software package)
Partner	Partner No = CN, Partner Name = Cname.	Client (ADABAS)
Partner Address	Partner No = CN, Address = Address.	Client (ADABAS)

Note that un-normalized cases will appear on the same table/file implementing more than one entity. I.e. The most badly structured legacy files can always be cross-referenced to the entity model provided enough cross references are used to show the convolutions.

The partitioned or replicated cases will appear on the same entity, being implemented by more than one table/file. For instance, a software package may duplicate some data, which is also maintained separately by other systems.

Application System	Table or File Name
Sales	Client (Oracle)
Contact Management	Customer (software package)
Shipping	Client (ADABAS)
Shipping	Shipment (ADABAS)

Dependence of Services on Repository Content

	CONTENT							
	An Entity Model	Business Process Models	Business Rules	Cross references between Entities and Processes	Cross references between Entities and the actual Tables/Files	Cross-references between the attributes of the entities and the columns/fields of the tables/files.	Cross references between the Processes and the Applications	Cross-references between the Applications and the Tables/Files.
Necessities for a service to either use or create and examine, specific contents. On a scale of 0 to 10, 0 means irrelevant, 10 means essential. (This is clearly a 'ballpark' estimate.)								
Selection of a software package	10	6	10	6	10	10	8	10
Development of a new application	10	6-10	10	6-10	10	10	8	8
Modification of an existing application	10	6	10	6-10	10	10	6	6

Inclusion of existing (legacy) applications	10	5	6	6	10	10	2	10
Inclusion of existing software packages	10	5	6	6	10	10	2	10
Business Re-engineering	10	10	10	10	0	0	8	0
Creation of Data Warehouses	10	6	10	6	10	10	0	2
Ongoing maintenance of Data Warehouses	10	6	10	6	10	10	0	2
Discovery, documentation and control of distributed and replicated data	10	6	10	6	10	10	2	6
Maintenance of integrity	10	0	10	0	10	10	0	0
Control and recording of business meaning	10	2	10	2	0	0	0	0
Migration of systems from present architecture to the desired architecture.	10	6-10	10	6-10	10	5	6	6

Interpretation of the State of the Repository.

The following two pages show one way to measure the repository quality and suggest how to interpret the situation and the effort needed for the different combinations of good and bad EntityMapping%, Table% and TableMapping%.

Note that the concept of Entity% has not been considered to be useful. It could be defined as the number of entities modelled in the corporate logical data resource (the fully detailed set of entity models that is **supposed** to cover the entire business) as a % of the number of entities **actually** needed to cover the entire business. It is a measure of the completeness of the entity model in terms of some ideal business.

The analysis demonstrates that the mapping between the entity model and the actual existing databases is an essential component of the metadata.

When an organization is experiencing a period of undisciplined evolution, the mappings can play a crucial role.

As new databases are added (sometimes in an ad hoc manner), whether through acquisition of software packages or through in-house development, they can be reverse engineered into the entity models, and the mapping between them and the entity models, documented. In the process, anomalies, inconsistencies and duplications resulting from the ad hoc acquisitions, can be made evident through the mappings.

Measures of Repository Quality		Metadata in the Repository
Metadata about existing Databases <ul style="list-style-type: none"> Table_% (% of existing tables/record types represented in repository) Column_% (% of existing columns/fields represented in repository) Key_% (% of existing table/record type keys/unique descriptors represented in repository) ForeignKey_% (% of existing foreign keys/descriptors represented in repository)	↔	Metadata about existing databases. <ul style="list-style-type: none"> Tables (columns, keys, foreign keys, location, database platform, ...) ADABAS record types (fields, unique descriptors, 'foreign descriptors', location, ...) Etc
Mapping <ul style="list-style-type: none"> EntityMapping_% (% of repository entities mapped to one or more repository tables/record types) AttributeMapping_% (% of repository entities mapped to one or more repository tables/record 	↔	Mapping between entity models and databases. <ul style="list-style-type: none"> Entities = Tables/Record types (the same entity could be mapped to more than one table and record type; the same table/record type could be mapped to more than one entity if not normalized - at least one key of the entity must map to at least one key of the table/record

<ul style="list-style-type: none"> types) <ul style="list-style-type: none"> RelationshipMapping_% (?) TableMapping_% (% of repository tables/record types mapped to entities) ColumnMapping_% (% of repository columns/fields mapped to attributes) <p>Confidence Level that the semantics of the existing database elements is consistent with the semantics of the corresponding entity model elements (in the case where there is a mapping).</p>		<ul style="list-style-type: none"> type) <ul style="list-style-type: none"> Attributes = Columns/Fields (the same attribute could be mapped to more than one column/field - foreign keys or replication) <p>Relationships = Foreign Keys etc</p>
<p>Metadata about business meaning</p> <ul style="list-style-type: none"> Confidence Level that the semantics of the entity model actually fits the business reality (correct, complete, stable, simple). 	↔	<p>Metadata about the business meaning & business rules of all the business data of the organization.</p> <p>Entity Model (entities, attributes, unique-ids, relationships, supertypes, subtypes, subject areas, ...)</p>
<p>Note: The word `existing` means actually physically present, being used by some system, at any site in the organization.</p>		

Interpretation of the State of the Repository (A good state is shown by ♥)			
		Table_% is Good	Table_% is Bad
EntityMapping_% is Good	TableMapping_% is Good	Most of the Entity Model is implemented and most of the databases are explained. ♥	<i>Effort is needed</i> to record the many missing databases and to add the corresponding entities and mappings.
	TableMapping_% is Bad	Most of the Entity Model is implemented but <i>effort is needed</i> to add the many missing entities for the recorded databases and map them to tables.	<i>Effort is needed</i> to record the many missing databases and to add the corresponding entities, as well as add missing entities for databases already recorded, and the mappings.
EntityMapping_% is Bad	TableMapping_% is Good	Most of the Entity Model is un implemented but most of the databases are explained. A large part of the Entity Model represents future needs. The entity model is close to complete in terms of present databases. ♥	<i>Effort is needed</i> to record the databases into the repository and map these new tables to entities.
	TableMapping_% is Bad	Most of the Entity Model is un implemented and most of the databases are un explained. <i>Effort is needed</i> to map entities already in the repository to tables already in the repository.	<i>Effort is needed</i> to record the databases into the repository and map these new tables to entities, as well as the tables already in the repository to entities.