Notes on the transformation of the CIDOC relational data model

1. Introduction

The present document is a relatively informal collection of remarks and comments arising from work on the conversion of the CIDOC relational data model at ICS FORTH in July 1996. The following people took part:
Martin Doerr
Nicholas Crofts
Ifigenia Dionissiadu
Christina Gritzapi

The primary goal of the current work on the transformation of the CIDOC relational model is to demonstrate the feasibility of adopting an OO approach. The intended audience is the CIDOC documentation standards work group. This document is not intended for distribution to the wider CIDOC community

2. Scope and role of the draft model

2.1 *Interpretation of the existing model*

A straightforward 'mechanical' transformation of the existing model is not very illuminating. The main reason for this is that the OO model has the potential for a higher level of semantic content and a greater degree of precision than a relational model. Some interpretation of the existing model is therefore needed in order to construct a meaningful OO model. Some of the necessary information is contained in textual notes and comments in the documentation associated with the relational model, but this is not always sufficient. Data, especially authority lists, are often needed as well Inevitably this leads to the possibility of misinterpretation. The problem is analogous to having a construction kit without all the instructions for putting it together.

The draft OO model will have to be controlled and verified by domain experts in order to ensure that it reflects the intentions underlying the original relational model.

2.2 *Role of the reference model*

The reference model is not intended primarily as a basis for implementation. We see the role of the reference model as being to define a semantic framework which will enable compatible systems to exchange and share information. (Information exchange includes issuing queries over the net and receiving answers from heterogeneous sources.) This represents something of a paradigm shift with respect to the existing data model.

Many formats are currently available which allow relatively simple are unambiguous exchange of data, however, the meaning of these data, their scope and application, is often far from obvious. The OO reference model provides a means for defining the semantic value of data and thereby facilitates information exchange.

2.3 *Granularity*

The current draft version of the OO reference model is limited in detail to what it was possible to achieve in the time available. We have intentionally restricted our attention to the primary entities and the more ‘interesting’ relations. A substantial amount of detail needs to be added to the model.

Another issue is the degree of granularity which the reference model *should* contain. This needs to be sufficient to ensure semantic compatibility between different realisations but should not be restrictive. At present, the definition of the required level of granularity would seem to be intuitive and depends on a number of factors, including the quantity of data to be handled. However it may be possible to formulate some more precise guidelines.

The question of granularity needs to be considered from different angles: the depth required by specific applications (which will extend the class hierarchy to incorporate domain specific details) and the depth and level of complexity at which the hierarchy becomes unmanageable. Extended causality chains or part

interconnections, e.g. book - print - print-stock - stock creation - author, may have to be avoided if they apply only to a few cases.

2.4 *Compatibility*

We have assessed the possibility of reducing the an OO model to a Relational equivalent. This does not appear to present any major conceptual difficulties though we have not attempted to define the formal rules which should be applied.

2.5 *Terminology*

Much of the terminology used by computer scientists is not standardised hence difficult to understand - even for computer scientists! This aggravates the problem of communicating with non specialists. The term ‘Data model’ is a case in point. Computer scientists generally use this term to refer to different modelling schemas: the OO data model, or the relational data model. This is contrary to current CIDOC usage which employs the term as a contraction of ‘the model of the data’.

In view of the general acceptance, within CIDOC, of this latter use of ‘data model’ it seems clear that we should continue to use this term as at present, adding a note for computer scientists to the effect that the term is not used as they might expect.

2.6 *Role of the meta-model*

The CIDOC OO reference model contains a semantic meta-model. An important role of this model is to define semantic extension rules. Typically, this specifies structural constraints on the sorts of links and subclasses which can be created: the notion of ‘style’, for example, could be restricted to man-made objects. The meta model also provides a means for 'talking about the model'. Classes and attributes are grouped together into intuitive categories: e.g. spatial properties, temporal properties, etc.

NB systems based on the formal language TELOS, as the SIS, product of FORTH, allow the physical implementation of meta-models, but this is by no means essential for the reference model to be used.

3. Formal considerations

3.1 *Documents*

We propose to present the following documents.

	Title	*Audience*	*Authors*
1	Notes on the transformation of the CIDOC relational data model.	Data standards work group (internal)	NC/MD/ID/CG
2	Formal transformation rules and principles used in transforming the model	Computer scientists, Data standards work group	MD/CG/NC
3	Introduction to the CIDOC reference model • Goals and objectives of reference model • Introduction to OO • Simplified class hierarchy	CIDOC	NC/ID/MD/JS/ PR
4	CIDOC OO reference model • OO data model[1] • OOo modelling principles (structural meta-model) • Extension rules (semantic meta model) • Implementation rules	CIDOC	

The current document (1) forms the basis for preparation of documents 2 and 3. Document 2 is essentially a technical paper intended for computer scientists and members of the Documentation standards work group. However, document 3, Introduction to the reference model, is intended for a much wider audience. Our goal is to have this latter document ready for Nairobi. This document will contain (at least) a definition of the goals of the reference model, a non-technical explanation of the OO approach, and a simplified presentation of the OO class hierarchy, accessible to non experts. This class hierarchy will need to be updated on a regular basis.

Document 4 is the reference model itself. Elaborating and defining this model will require a considerable amount of effort. Maintaining it will constitute an ongoing task for the Documentation standards work group.

3.2 *Acceptance*

Formal approval of the propositions contained in the current document will need to be given by the Documentation Standards Work Group, preferably at the next meeting in Nairobi.

3.3 *Modelling tools*

The initial modelling process was conducted using SIS tools, products of FORTH. In the future we intend to adopt ISO standard 10303-11, known as EXPRESS-G, to represent the data model diagrams. This standard is supported by a number of editing tools from various software producers world-wide.

A simplified class hierarchy will also be presented in HTML format for easy consultation via Internet. This presentation is language-neutral and machine-readable.

In order to maintain a high degree of compatibility we have used a subset of current OO dialects.

3.4 *Naming conventions*

Entities and attributes taken more-or-less directly from the existing relational model are in upper-case, as at present. New classes and attributes are in lower case. In choosing attribute names we have tried to remain close to natural language. Attributes for which there is no intuitively obvious label are prefixed with 'has_'.

[1] The term 'data model' signifies 'the model of the data', and differs from the sense generally recognised by computer scientists.

NB Attribute names need to be read in the context of the class which contains them.

3.5 *Ownership*

The CIDOC OO reference model and related documents are the property of CIDOC.
Several issues need to be clarified:

Copyright
Intellectual property
Authorship and credits
Diffusion rights
Modification rights
Approval mechanism

ICS-FORTH wishes to include a 'no mutual claims' clause on the conceptual contents, to ensure that they will be able to continue to benefit from and to make use of their ideas incorporated in the OO reference model.

4. Methodology and problems

4.1 *Primary and foreign keys*

The use of the terms ‘primary key’ and ‘foreign key’ appears inconsistent in the documentation associated with the current relational model. Consequently we have not used this information.

4.2 *Methods and constraints*

An important aspect of the OO approach is the possibility associating objects with ‘methods’ which encapsulate their behaviour. We have decided not to include method definitions in the reference model for several reasons:

1. The information necessary for the creation of such methods is currently contained in complementary documents such as the minimal data standard in the form of recommendations for business rules and database integrity constraints. These documents need to be consolidated and their precise relation to relational model clarified.
2. Current OO technology does not provide a uniform and consistent means of documenting and implementing object methods.
3. It is unclear whether the use of methods to instantiate constraints in desirable. Encapsulation may prove to be a handicap in the current context.
4. Constraints are often function specific and related to a particular implementation. As such they do not belong in the reference model.
5. On some occasions, data may be inconsistent - as when two sources conflict. Constraints may force the unwanted resolution of such inconsistencies.

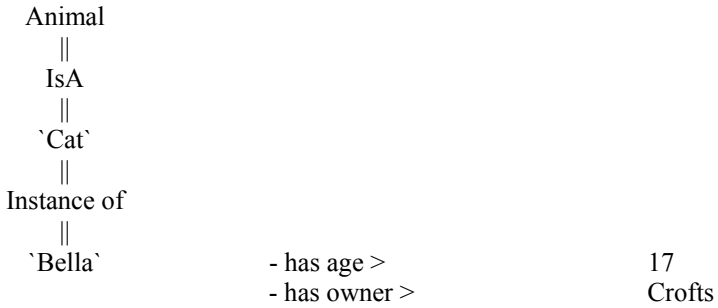
4.3 *Need for data*

In the conversion process, some *data* elements contained in a relational model may become *structural* elements such as classes and links. This is typically the case with authority lists or terminological data.

In the following example, data from table Animals is modelled as an instance ‘Bella’ of class ‘cat’ with the attributes age 17 and owner Crofts. Cat is a subclass of Animal. Note that the owner attribute of cat should be a link to an instance of class ‘person’.

Animals

name	type	age	owner
Bella	cat	17	Crofts



As it stands, some ‘structural’ data are missing from the relational data model. One notable example is the OBJECT & EVENT intersection entity which does not define any relation types. This could lead to divergent and incompatible realisations based on the same relational structure.

The OO approach requires these data to be defined as part of the model and consequently the possibility of creating incompatible systems is reduced.

The absence of these ‘structural’ data limits the level of detail which it is possible to include in the draft OO class hierarchy. We have taken the option of inventing examples of some the ‘missing’ data.

e.g The OBJECT & EVENT entity should at least define relation types such as ‘Created’, ‘Destroyed’, ‘Born’, ‘Died’, etc.

4.4 *Normalisation of the relational model*

Close examination of the current relational model revealed some de-normalised elements: redundant fields, missing fields, inconsistent data structures, etc. We took the treatment of dates as an interesting example. Currently date fields are contained within many entities and their treatment is not always consistent. We have attempted to normalise the data structures as part of the conversion process.

All dates are now unified by a date class: TIME-SPAN, which contains upper and lower limit dates, display format, etc. This ensures consistent handling of dates. We have not gone so far as to include the complex processing rules which are involved in handling dates.

4.5 *Interpretation of ambiguous structures*

Certain relationships in the existing model appear unclear or incomplete. This is the case for example in the complex relationship between Event, Time-span and Place. The place attribute is not present in the TIME-SPAN entity. The EVENT entity allows for multiple time-spans and multiple places, but there is no bonding between them. EVENTS are not clearly linked to the agents who are responsible for them.

The draft model proposes the creation of a 'Period' super class, which has both time and place attributes. 'Events' are a subclass of Period. 'Method_use' events are a subclass of events. This hierarchy provides a precise means of binding places, people and events.

e.g A sandwich is created. The creation is a type of Event and we can specify where and when the event took place: Crete, 1996. The event falls within an historical period (Modern Greece) which has its own time and place: 1823 - present day, Greece.

4.6 *Representation semiotics*

The notion of representation in images presents an interesting problem. Objects or people represented in a picture may be real, in which case they are instances of classes, or they may simply be imaginary. Imaginary objects do not usually exist in a database as instances in the same way as real objets. One possible solution would be to create links to the appropriate classes, thus a picture of a cow would be associated with the class ‘cow’ and not with any particular cow. Unfortunately, most OO systems do not allow this. Creating an instance of an imaginary cow for each picture of a cow would be possible, but confusing in practice. As a solution we propose the creation of a generic instance of an ‘archetype’ cow to which pictures of cows could refer.

4.7 *Events, Methods and roles.*

The OO approach allows a considerable simplification of the complex relation between events, methods and roles without any loss of semantic content. The existing METHOD entity is replaced by the ‘Method’ class, which is a sub class of Event. Method is in fact an example of an *abstract* class since it would not normally have any instances. ‘Method use’ is an instance of Method and documents an occasion on which a particular technique or method was employed. Method use inherits time-span and place attributes from event. It also has attributes Agent, and Object: the person who was responsible for employing the method and the object which was affected. The additional attribute ‘consists of’ allows complex processes consisting of several methods use events to be documented.

As in the current relational model, no attempt has been made to model the logic by which events may be related to create processes.