Four-level Architecture for Closure in Interoperability

Nick Rossiter[1] and Michael Heather[2]

[1] School of Informatics, Northumbria University, NE1 8ST, UK,
nick.rossiter@unn.ac.uk,
WWW home page: http://computing.unn.ac.uk/staff/CGNR1/
[2] Sutherland Building, Northumbria University, NE1 8ST, UK

Abstract. A definition of types in an information system is given from real-world abstractions through data constructs, schema and definitions to physical data values. Category theory suggests that four levels are sufficient to provide ultimate closure for computational types to construct information systems. The Godement calculus provides rules governing the composition of the mappings at different levels. Examples of information systems are reviewed in terms of the four-level architecture including IRDS, the Grid, the semantic web and MOF/MDA.

1 The four fundamental levels and their formalisation

Interoperability is still a major problem in information systems. Most achievements have been with systems using a similar model or paradigm. Where heterogeneous systems are involved, progress has required much manual adjustment to mappings. Recently the development of the Grid has exposed the great difficulty of employing data held in formal database systems as opposed to operating system files [Watson, 2002]. Using higher-order logic we build on existing work [Rossiter, Nelson and Heather, 2001] to review some examples and their reliability for applications of interoperability and cross-platform software.

The whole subject of relating different systems emerges in federated information systems (FIS) as the core issue. The term *level* is used in FIS in a subtly different way [Conrad *et al*, 1997] to that employed in this paper so we start with a brief description of our architecture. One instance of the fundamental levels in Fig 1 is a representation of a single platform, paradigm or model. Level 1 would be real-world type abstractions, level 2 the type constructs available, level 3 the data types and level 4 the named values.

Constructive mathematics attempts to develop logically what can work in practice and can therefore provide the necessary universal typing for interoperability of heterogeneous data systems with consistency and quality assurance in the real-world. Category theory [Barr and Wells, 1990] is particularly appropriate for modelling multi-level relationships for it is essentially concerned with links between objects. In categorial terms each of the four levels is defined as a category (i.e. a type) as shown in Fig 2. Between each level there is a higher-order function, a functor, which ensures that certain consistency requirements

are met in the mapping between the source and target categories. The four levels from the top are defined as the categories **CONCEPTS** (abstractions), **CONSTRUCTS**, **SCHEMA** and **DATA** with the mappings between them as shown in the diagram.

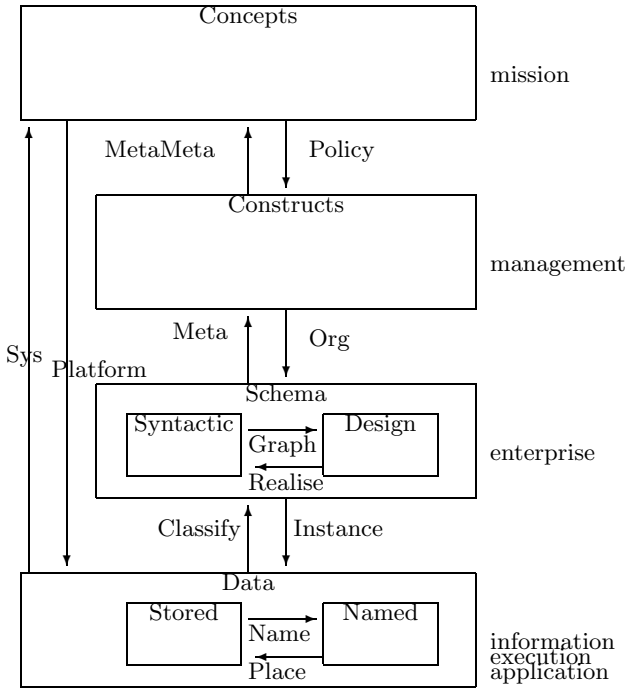

Fig. 1. Interpretation of Fundamental Levels informally

Adjointness [Barr and Wells, 1990] characterises the unique relationship between a lower-limit functor (F) that preserves co-limits and an upper-limit functor (G) which preserves limits, written $F \dashv G$, that is G is right-adjoint to F. The multi-level application shown in Fig 2 involves the composition of adjoints, that is an expression is derived in which two or more adjoints are adjacent to each other. It is part of the power of category theory that adjoints can be composed in the same way as other arrows. For example consider the adjoints shown in Fig 3 where **CC** is the category **CONCEPTS**, **CS CONSTRUCTS**, **SM SCHEMA** and **DT DATA**.

Then we may have six adjoints (if the conditions are satisfied):
$I \dashv C$; $O \dashv M$; $P \dashv A$; $I \circ O \dashv M \circ C$; $O \circ P \dashv A \circ M$; $I \circ O \circ P \dashv A \circ M \circ C$
where P is the functor $Policy$, O Org, I $Instance$, A $MetaMeta$, M $Meta$ and C $Classify$. We can construct the 4-tuple to represent the composed adjunctions defined in Fig 2: $< IOP, AMC, AM\bar{\bar{\eta}}_{cc}OP \bullet A\bar{\eta}_{cc}P \bullet \eta_{cc}, \bar{\bar{\epsilon}}_{dt} \bullet I\bar{\epsilon}_{dt}C \bullet IO\epsilon_{dt}MC >$.

If the conditions of this adjunction are met, we can represent the composed adjunction $Platform \dashv Sys$ by the 4-tuple $< Platform, Sys, \eta_{cc}, \epsilon_{dt} >$: **CC** $\longrightarrow$

DT where $Platform = IOP$, $Sys = AMC$, η_{cc} is the unit of adjunction, ϵ_{dt} is the counit of adjunction, cc is an object in **CC** and dt an object in **DT**.

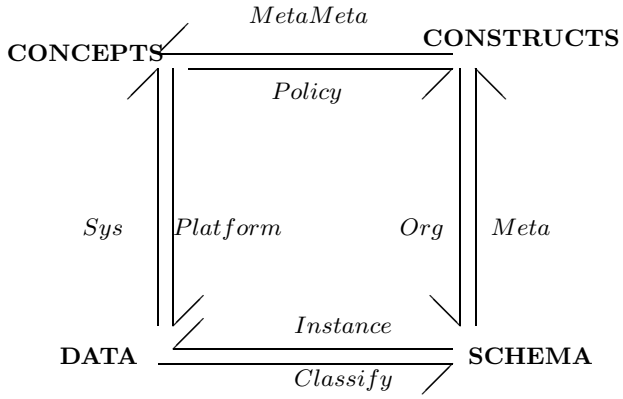

Fig. 2. Four Levels in Functorial Terms

This adjunction can be evaluated for each application giving a collection of 4-tuples. Comparison of these 4-tuples then gives the mechanism for computational type closure. The ability to compose adjoints naturally means that we can combine well together such diverse features as policy, organization and data in a single arrow.

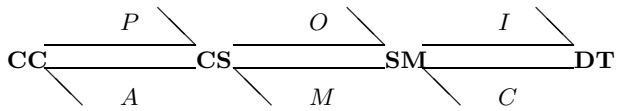

Fig. 3. Composition of Adjoints

The overall composition gives a simple representation for conceptual purposes; the individual mappings enable the transformations to be followed in detail at each stage and provide a route for implementation.

2 Comparing one System with Another

Adjunctions give the relationships between one level and another. We can also approach the problem by considering a direct mapping between one instance of the four-level architecture and another as in Fig 4. Here for simplicity the mappings are viewed in one direction only. Two systems are compared, one involving categories **CC**, **CS**, **SM** and **DT**, the other **CC**, **CS′**, **SM′** and **DT′**. **CC** is the same in both systems as there is one universal type for concepts. As in Fig 3, the functors relate the categories. We have now though added natural transformations to relate the mapping between one functor and another.

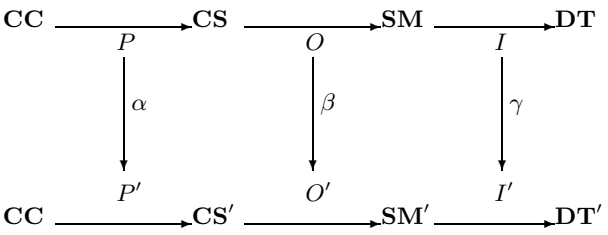

Fig. 4. Comparison of Mappings in two Systems

If we follow the constructive principles of category theory, then the composition of these arrows is natural. The Godement calculus ([Godement, 1958]; [Barr and Wells, 1990], pp 94-97) gives a number of rules governing the compositions. Rules G2 and G3 say that the composition of functors and natural transformations is associative so that for instance:
$(I'O')\alpha = I'(O'\alpha)$; $\gamma(OP) = (\gamma O)P$
Rule G3 says that natural transformations may be composed with each other:
$\gamma\beta = (\gamma O) \circ (I'\beta)$; $\beta\alpha = (\beta P) \circ (O'\alpha)$

The consequence of this for interoperability is that a categorical approach ensures that the various arrows of different types can be composed with each other, irrespective of their level in the system. Equations can be derived, representing an equality of paths, with unknown components that can be determined from an evaluation of the known properties. For instance with the path IOP from $\mathbf{CC} \longrightarrow \mathbf{CS} \longrightarrow \mathbf{SM} \longrightarrow \mathbf{DT}$ defining an object-oriented system, then the path $I'O'\alpha$ from $\mathbf{CC} \longrightarrow \mathbf{CS}' \longrightarrow \mathbf{SM}' \longrightarrow \mathbf{DT}'$ would define a relational representation if P' maps onto relational constructs in the category $\mathbf{CS}'$.

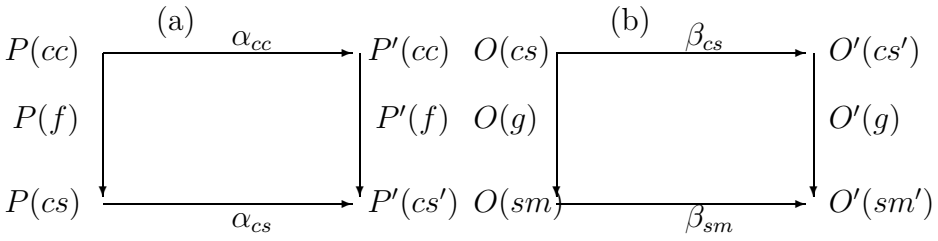

Fig. 5. Commuting Target Square for Natural Transformations: (a) $\alpha : P \longrightarrow P'$, comparing policies; (b) $\beta : O \longrightarrow O'$, comparing use of constructs

In category theory four levels are required to define an arrow as unique up to natural isomorphism. The four levels are: 1) object or identity arrow (within a category), 2) category (comparing objects), 3) functor (comparing categories) and 4) natural transformation (comparing functors). No more levels are required. An arrow comparing natural transformations is a natural transformation. Two squares, derived from Fig 4, are shown above. Fig 5(a) must commute for each arrow $f : cc \longrightarrow cs$ if α is to be a natural transformation. Similarly Fig 5(b) must commute for each arrow $g : cs \longrightarrow sm$ if β is to be a natural transformation. Viewed in this way a natural transformation is not a layer above functors and functions. The levels are interwoven with natural transformations considering how every arrow defined at the lowest level is mapped.

If we write the arrow: $\delta : \alpha \longrightarrow \beta$ then δ is a composition $\beta \circ \alpha$. An arrow from one natural transformation to another gives a composition of the natural transformations, not a new level ([Barr and Wells, 1990], at p.85). The four levels of concepts, constructs, schema and data are viewed in Fig 2 as four categories connected by a composition of three functors. An alternative view, shown in Fig 6, is closer to the four levels inherent in category theory. The fundamental levels are considered to be data values, named values, classified values and contrasted representation corresponding in category theory to object, category, functor and natural transformation respectively. The natural transformations are now the duals of those shown earlier in Fig 4 as indicated by the * superscript.

alternative fundamental levels	category theory levels	four levels of Fig 4
1. data values	objects (identity arrows)	id_{dt}
2. named values	category	**DT**
3. classified values	functor	C : **DT** $\longrightarrow$ **SM**
4. contrasted representation	natural transformation	$\alpha^* \circ \beta^*$

Fig. 6. Alternative Interpretation of Levels in the Architecture

This view does not supersede the earlier one which is more useful for system design with its similarity to the ANSI/SPARC three-level architecture. The alternative view though may have some potential for interoperability where comparisons are an inherent part of the methodology with natural transformations as ultimate closure. It can be seen that the addition of further levels is possible but nothing is gained by it type-wise. Thus addition of an extra level to the top of Fig 1 simply results in the top level being a composition of three arrows rather than two. The practical consequence is that a fifth level is equivalent to an alternative fourth level. The meta-meta level gives ultimate closure of types.

3 Levels in applications

Four existing approaches to interoperability were examined to see how they compare in giving a genuine four-level strategy for tackling the problem of interoperability [Rossiter, Nelson and Heather, 2003]:

1. Information Resource Dictionary System [Gradwell, 1990]
2. Grid Construction [Watson, 2002]
3. Data Exchange Languages [Berners-Lee *et al*, 2001]
4. Metaobjects in the Model-driven Architecture [Bezivin, 2001]

The Information Resource Dictionary Standard (IRDS) did provide four levels but ANSI downgraded this standard and its influence has been less than anticipated. More recently ISO has begun again to value a four-level architecture with the consideration of a meta-meta *model* in work on comparing models. Systems developed recently, claiming to provide interoperability, such as MOF are able to provide considerable assistance within a paradigm but appear to lack the top level, mapping abstractions to constructions, necessary to achieve interoperability across paradigms. Moreover, recent work has suggested that future effort with MOF [Habela, Roantree and Subieta, 2002] should flatten the metamodel to reduce complexity and to support extendibility. Both IRDS and MOF are data-driven approaches in a general sense. The semantic web takes a different approach, being partially data-driven through RDF but also relying on agent-based technology for resolving mismatches. The semantic web therefore appears to lack the two top levels of concepts and constructs but the use of ontologies and agents may compensate to some extent at least for some of this loss. The Grid also lacks the top two levels for data addressing and its potential will not be realised until this deficiency is tackled.

To conclude the generality of current techniques for interoperability is in doubt. The definition of the four levels necessary for providing interoperability, the availability of the Godement calculus for composing mappings formed at different levels and the specifications of the adjointness between the levels and of pullback categories representing relationships, all add coherence through a categorical approach to interoperability.

References

[Barr and Wells, 1990] Barr, M, & Wells, C, *Category Theory for Computing Science.*

[Berners-Lee *et al*, 2001] Berners-Lee, T, Hendler, J, & Lassila, O, The Semantic Web, *Scientific American*, May 2001.

[Bezivin, 2001] Bezivin, J, From Object Composition to Model Transformation with the MDA, *3rd ICEIS*, Setubal, invited paper.

[Conrad *et al*, 1997] Conrad, S, Eaglestone, B, Hasselbring, W, Roantree, M, Saltor, F, Schnhoff, M, Strssler, M, & Vermeer, M W W, Research Issues in Federated Database Systems: Report EFDBS '97 Workshop, *SIGMOD Rec* **26**(4) 54-56.

[Godement, 1958] Godement, R, *Théorie des faisceaux*, Hermann.

[Gradwell, 1990] Gradwell, D J L, The Arrival of IRDS Standards, *8th BNCOD*, York 1990, Pitman 196-209.

[Habela, Roantree and Subieta, 2002] Habela, P, Roantree, M, & Subieta, K, Flattening the Metamodel for Object Databases, *ADBIS 2002*, 263-276.

[Rossiter, Nelson and Heather, 2001] Rossiter, B N, Nelson, D A, & Heather, M A, A Universal Technique for Relating Heterogeneous Data Models, *3rd ICEIS*, Setúbal, I 96-103.

[Rossiter, Nelson and Heather, 2003] Rossiter, N, Nelson, D A, & Heather, M A, Formalizing Types with Ultimate Closure for Middleware Tools in Information Systems Engineering, 5th ICEIS, Angers, France 23-26 April 8pp.

[Watson, 2002] Watson, P, *Databases and the Grid*, Computing Science Tech Rep no.755, University of Newcastle upon Tyne 16pp.