ENHANCING THE LINKING OF SOFT SYSTEMS AND INFORMATION SYSTEMS MODELS THROUGH A UNIFORM AND FORMALISED REPRESENTATION

Quan C. Dang
School of Informatics and Multimedia Technology
University of North London
2-16 Eden Grove, London N7 8EA, England

ABSTRACT

There are numerous works suggesting linking SSM models to information systems (IS) models for information systems development. In these works links between SSM and IS models are established at the conceptual level i.e. they provide ways to utilise SSM models to identify or to derive IS models. However, most of them have not provided a method for representing the links explicitly in the combined model. Consequently, the links become subtle and untraceable, looking at the combined model only at the representation level.

This paper proposes an approach based on meta modelling to achieve a uniform and formalised representation of models that combine SSM and IS models. The approach allows one to represent not only SSM and IS modelling concepts but also relationships between the concepts in a meta model using the conceptual modelling language Telos. Telos supports meta models to be flexibly specified and extended to meet specific modelling requirements of a particular project.

An exemplary application is presented to demonstrate the operationalisation of the proposed approach and to illustrate the uniform representation of combined models. The example also shows benefits of the formalised representation in terms of computer support for managing and retrieving a combined model's meta data.

INTRODUCTION

It is well-recognised that the appreciation of the problem situation, in particular its relevant human activities, plays a crucial role in successful development of information systems, see e.g. Checkland (1995), Avison & Wood-Harper (1990) and Stowell (1995). Given this, Soft Systems methodology (SSM) (Checkland, 1981) is advocated to provide the IS analyst with a powerful methodology to obtain a rich understanding of the problem situation for which an information system is built (Checkland, 1995; Stowell, 1995). Effectively, applications of SSM in information systems development (ISD) range from the planning of IS strategies (Galliers, 1992), IS requirement analysis (Stowell & West, 1994), IS analysis and design (Avison & Wood-Harper, 1990), to evaluation of IS methodologies (Jayaratna, 1994).

In this paper, we focus on the use of SSM in the context of information systems modelling (Avison & Fitzgerald, 1995; Flynn & Diaz, 1996; Loucopoulos & Zicari, 1992). Specifically, we are concerned with the fashion in which soft systems models, and/or knowledge gained from soft systems analysis, are linked to information models, such as the data/structural model and process/behavioural model. For a review of these models, see e.g. Avison & Fitzgerald (1995).

We begin by discussing issues inherent to the linkage of SSM to information systems modelling and drawbacks of the current approaches. Subsequently, we propose an approach to overcome the drawbacks, based on meta modelling techniques; the language Telos utilised to implement the approach and a method for constructing a meta model that unifies SSM and information models are outlined. This is followed by a section that presents an exemplary application of the method to build a meta model. Benefits of using the meta model in modelling problem domains are highlighted. The final section gives an appraisal and discusses relationship of the proposed approach to other work.

BACKGROUND: LINKING SSM AND INFORMATION SYSTEMS MODELS

In soft systems practice, work that utilised SSM in helping information systems development was reported as early as in 1970 (Checkland & Griffin, 1970). Following the publication of Checkland (1981) a great many papers on using SSM in ISD appeared, see e.g. Stowell (1992), Avison & Fitzgerald (1993), Stowell (1995). According to Checkland (1995), these approaches can be classified into two main streams. The first stream involves linking SSM to well-established information systems design methods, e.g. CCTA (1991), Prior (1990), and Savage & Mingers (1996). The second stream explores SSM in creating an information strategy in an organisation, e.g. Galliers (1992). The work in both streams faces the problems of making links between SSM's models and the data-oriented considerations or information systems models (hereafter called information models for short). The issues and problems related to linking SSM to ISD are highlighted and discussed in numerous publications, from both methodological and technical viewpoints. Detailed discussions can be found in, for instance, Checkland (1992), Miles (1988), Mingers (1995) and Checkland & Howell (1998).

This research has observed that none among the current approaches has provided a model combining SSM's and IS models which has a uniform formalised representation. For a review of these approaches see e.g. Mingers (1995). In fact, the issue of uniform and formalised representation of the combined model has not been addressed by most of the approaches. In Gregory & Merali (1993), a method for representing SSM conceptual models using Prolog is described. However, this method does not address the general issue of the representation of the knowledge captured by soft system analysis. Specifically, the knowledge captured by the rich picture is not included in the final Gregory & Merali's model. In other approaches, such as CCTA (1991), Savage & Mingers (1996) and Avison & Wood-Harper (1990), it is not unusual for combined models to be represented using several languages. These approaches just provide frameworks in which different (sub-)models are conceptually combined/linked, but the original languages of the constituting sub-models are still utilised as such. This results in the following drawbacks:

- the links between the constituting models are not explicitly present in the final model, though they may be stored in the modeller's head or documented in an informal format;
- the resultant models, or the knowledge captured by the modelling, are difficult to manage. Because the models are represented in different formats, which may be both formal and informal, there is no formalised way to impose and check the model in terms of its consistency and (meta-)data redundancy over the entire model;
- modelling activities are difficult to support by a software tool.

These drawbacks may be overcome with a uniform and formalised representation of the combined model, provided that such a representation is capable of representing aspects captured by soft systems and information models as well as links between the aspects. In addition, the representation formalism used should not impose any extra structure on the soft systems modelling, in order not to constrict the exploratory nature of soft systems enquiry. The SSM incremental modelling process should also be supported when using the formalism.

In the next section, an approach that implements these ideas using the conceptual modelling language Telos (Mylopoulos *et al,* 1990) is presented.

A META MODELLING APPROACH

Meta modelling

Meta modelling is an activity of building a model of a model (Kottemann & Konsynski, 1984; Mylopoulos *et al*, 1990; Jeusfeld *et al*, 1997). It allows one to specify the semantics of, and structural relationships between, the modelling concepts of a model. The resultant meta model can then be exploited as a modelling language to represent the real world abstracted using the model.

A meta model allows modelling languages to be specified in a meta modelling language, which facilitates the interoperability and communication among the constituting models (Loucopoulos & Zicari, 1992; Jeusfeld *et al*, 1997; Avison & Fitzgerald, 1995). On the basis of a meta model, a CASE tool can be built to support modelling activities, maintaining the consistency and integrity constraints of the overall model.

In order to achieve a uniform and formalised representation of the SSM models and information models we have utilised meta modelling techniques. Our approach is to represent the set of modelling concepts of both SSM and information models along with their relationships using a meta modelling language. This language should meet the requirements indicated in the previous section. It is worth noting that the application of the meta modelling techniques is not straightforward because there is no universal meta modelling language. Most of existing meta modelling languages have been developed to support a pre-determined set of models, modelling formal and structured knowledge (Jeusfeld *et al*, 1997; Loucopoulos & Zicari, 1992). The language to implement our approach needs to be capable of representing both formal and informal knowledge and supporting incremental modelling.

The conceptual modelling language Telos and the ConceptBase system

Several languages have features that satisfy the indicated requirements, such as OMEGA (Attardi & Simi, 1981), CLASSIC (Brachman *et al*, 1992), and Telos (Mylopoulos *et al*, 1990). In comparison to OMEGA and CLASSIC, Telos is a conceptual modelling language that is specifically designed to

represent the knowledge of categories that pertain to information systems development. For more details see Mylopoulos *et al* (1990).
Telos is capable of representing informal knowledge that is represented in the natural language, as in soft systems models, because it implements the *semantic network* knowledge representation scheme, see e.g. Quinllian (1968). Besides, being an *object-centred* knowledge representation language, see e.g. Nilsson (1982), Telos supports various formal abstraction mechanisms such as *classification*, *generalisation-specialisation* and *aggregation* (Brodie *et al*, 1984). A *typed first order assertion sublanguage* is offered in Telos as a means of specifying *integrity constraints* and *deductive rules*.
The other reason for choosing Telos is that it is fully implemented in the ConceptBase system (Jarke, 1995 and Jeusfeld *et al*, 1997). ConceptBase is a deductive object base manager with a graphical user interface that allows Telos objects to be created, stored, retrieved, visualised and queried using the system in both textual and graphical formats.

Specification of modelling languages as meta models in Telos

A meta model of modelling languages can be constructed using the primitive objects of the Telos kernel object model (Jeusfeld *et al*, 1997). This kernel model can be extended with objects that define new modelling constructs. Structural and semantic constraints among the constructs are specified as Telos objects, assertions and abstraction mechanisms. The resultant meta model's collection of objects can then be employed as a modelling language to represent application domain models.
Given the expressive power of Telos, it can be used to build a meta model that is a modelling language that unifies SSM's models and information models.

OPERATIONALISATION OF THE APPROACH: AN EXEMPLARY APPLICATION

In this section, an exemplary application is presented in order to show the operationalisation of the proposed approach. The example demonstrates the use of Telos to build a meta model of a model that combines SSM models and information models. The meta model is then used as a unifying modelling language for modelling a problem domain, resulting in a model with uniform and formalised representation.
The SCORE model (Dang, 1997) is chosen for the purpose. However, it is worth noting that the presented method of building meta models is equally applicable to other models that combine SSM and information models.

A bird's eye view of the SCORE model

The SCORE model is designed with an aim to capture both human activity and information related aspects of an office environment in an integrated manner. In the model, an office environment is conceived as a problem situation with relevant human activities, and is modelled with soft systems. Resultant soft systems models are then enriched with events, entities, rules and constraints that are involved in the relevant activities. The constructs of event, entity, rule and constraint are deliberately chosen to capture the dynamic, structural and declarative knowledge of a problem domain respectively. These are principal bodies of knowledge required to be captured by an information model (Avison & Wood-Harper, 1990; Loucopoulos & Zicari, 1992; Avison & Fitzgerald, 1995 and Flynn & Diaz, 1996). Thus, the SCORE model represents the models that attempt to combine soft systems model and models of information modelling.

A meta model of the SCORE model

A meta model of the SCORE model has been built using Telos, for details see Dang (1997). The meta model defines all the modelling constructs of the SCORE model (namely, *rich picture, human activity system, conceptual model, entity, event, entity life cycle, rule and constraint*) and the links between the constructs in the form of *Telos object specifications*.
The meta model's specifications capture explicitly the semantics of the SCORE modelling constructs, which are of both soft systems and information modelling. This is illustrated by the object specification of the construct of *conceptual model* of a human activity system and the object specification of the construct of *entity life cycle*.
The specification of the object ConceptualModels (Figure 1) indicates that a conceptual model is built based on a root definition (see the attribute systemDefinition) and is composed of sub-activities linked

by logical links (see the attributes subActivity and logicalLink). Moreover, the specification requires that at least two sub-activities must be specified in a conceptual model (see the constraint containsTwoDifferentSubActivities).

```
Individual ConceptualModels in ScoreObjects,MetaClass,Class isA
SSMConstructs with
  attribute,necessary,single
     systemDefinition : RootDefinitions
  attribute,necessary
     subActivity : HumanActivitySystems;
     logicalLink : Links
  attribute
     monitorAndControl : HumanActivitySystems
  attribute,constraint
     containsTwoDifferentSubActivities : $ forall m/ConceptualModels
a1/HumanActivitySystems (m subActivity a1) ==> exists
a2/HumanActivitySystems ((m subActivity a2) and not (a1==a2)) $
end
```

Figure 1: The specification of the *conceptual model* construct

In comparison with the specification of the construct of conceptual model, the specification of the construct of entity life cycle is more complex (see Figure 2). In the SCORE model, the entity life cycle is represented using the *program structure diagram* (Jackson, 1983). The specification is composed of a collection of related objects, namely, EntityLifeCycles, ELC_elements, Events, NullEvents, Sequences, Iterations, Selections and TemporalLinks. The specifications of these objects embrace not only three basic control flow structures (i.e. *sequence*, *selection* and *iteration*) and temporal order of events of an entity life cycle, but also commandments that must be conformed to when constructing an entity life cycle. For instance, the commandment that cross-links are not allowed in an entity life cycle's tree (Jackson, 1983) is captured the constraint noCrossLink_in_a_lifecycle in the specification of EntityLifeCycles.

In addition to the semantics of individual modelling constructs, links between the constructs are explicitly embedded in the meta model. These links are (meta) modelled in several ways:

- by using attributes in specifications of modelling constructs, for example, the specification of the construct HumanActivitySystems includes the attributes involvedEntity, involvedEvent and governedBy, which indicate what entities, events, rules and constraints are involved in an activity;
- by defining aggregate objects, for example EntityLifeCycles and RichPictures;
- by defining special objects to represent particular types of relationships between modelling constructs. For example, the constructs Links, Boundaries, BinaryLinks and DirectedLinks are useful for representing lines/arcs, circles, and arrows between elements of soft systems rich pictures and conceptual models, as well as for representing specific relationships between arbitrary objects of a problem domain.

From the specification of the construct HumanActivitySystems given in Figure 3, it can also be seen that the meta model captures not only links that relate constructs belonging to an individual model (e.g. the *relevance* of a human activity to a *problem situation* and a *conceptual model* in a soft systems model), but also cross-model links that connect soft systems and information modelling constructs (e.g. the *involvement* of an *entity* in the execution of a *human activity*).

Furthermore, because Telos allows the meta-model to be adjusted and extended, new objects can be defined to meet specific modelling requirements, particularly to represent the knowledge that is not amenable to the pre-defined set of modelling constructs. This feature is useful, given the fact that the knowledge learned from a soft systems analysis is usually more than that is scripted using rich pictures, root definitions, conceptual models. In addition, using the ConceptBase system, the model's consistency and integrity are easier to maintain. That is, in a ConceptBase object base, deductive rules and integrity constraints defined in objects' specifications are automatically checked when a new object is added to the object base. Thus

```
Individual EntityLifeCycles in ScoreObjects,MetaClass,Class with
  attribute,necessary
     consistOf : ELC_elements
  attribute,constraint
     noCrossLink_in_a_lifecycle : $forall e/ELC_elements
sq1,sq2/Sequences it1,it2/Iterations sl1,sl2/Selections not(exists
elc1/EntityLifeCycles (elc1 consistOf e) and ((elc1 consistOf sq1)
and (elc1 consistOf it1) and (sq1 seq e) and (it1 iter e)) or
((elc1 consistOf sq1) and (elc1 consistOf sl1) and (sq1 seq e) and
(sl1 sel e))  or ((elc1 consistOf it1) and (elc1 consistOf sl1)
and (it1 iter e) and (sl1 sel e)) or ((elc1 consistOf sq1) and
(elc1 consistOf sq2) and (sq1 seq e) and (sq2 seq e) and
not(sq2==sq1)) or ((elc1 consistOf it1) and (elc1 consistOf it2)
and (it1 iter e) and (it2 iter e) and not(it2==it1)) or ((elc1
consistOf sl1) and (elc1 consistOf sl2) and (sl1 sel e) and (sl2
sel e) and not(sl2==sl1)))$
end
Individual ELC_elements in ScoreObjects,MetaClass,Class
end
Individual Events in ScoreObjects,MetaClass,Class isA ELC_elements
with
  attribute
     underlyingActivity : HumanActivitySystems;
     precondition : Proposition;
     participatingEntity : Entities;
     effect : Proposition
end
Individual NullEvents in ScoreObjects,MetaClass,Class isA
Events,ELC_elements
end
Individual Sequences in ScoreObjects,MetaClass,Class isA
ELC_elements with
  attribute,necessary
     seq : ELC_elements
  attribute
     timeOrder : TemporalLinks
  attribute,constraint
     containsTwoDifferentElements : $ forall s/Sequences
e1/ELC_elements (s seq e1) ==> exists e2/ELC_elements ((s seq e2)
and not (e1==e2)) $;
     needsTimeOrder : $ forall s/Sequences e/ELC_elements (s seq
e) ==> exists l/TemporalLinks (s timeOrder l) $
end
```

Figure 2: Telos objects necessary for representing the *entity life cycle* construct

```
Individual HumanActivitySystems in ScoreObjects,MetaClass,Class isA
SSMConstructs with
  attribute
     involvedEntity : Entities;
     involvedEvent : Events;
     governedBy : RulesConstraints;
     relevantTo : ProblemSituations;
     hasRD : RootDefinitions;
     hasCM : ConceptualModels
end
```

Figure 3: The specification of the *human activity system* construct

this supports soft systems incremental modelling. The example in Figure 4 shows a snapshot of a ConceptBase session in which an object that violates previously specified rules/constraints is rejected by the system when being added to the object base.

Once the meta model has been built, the modeller can employ the collection of the meta model's objects as a unified modelling language to model problem domains.

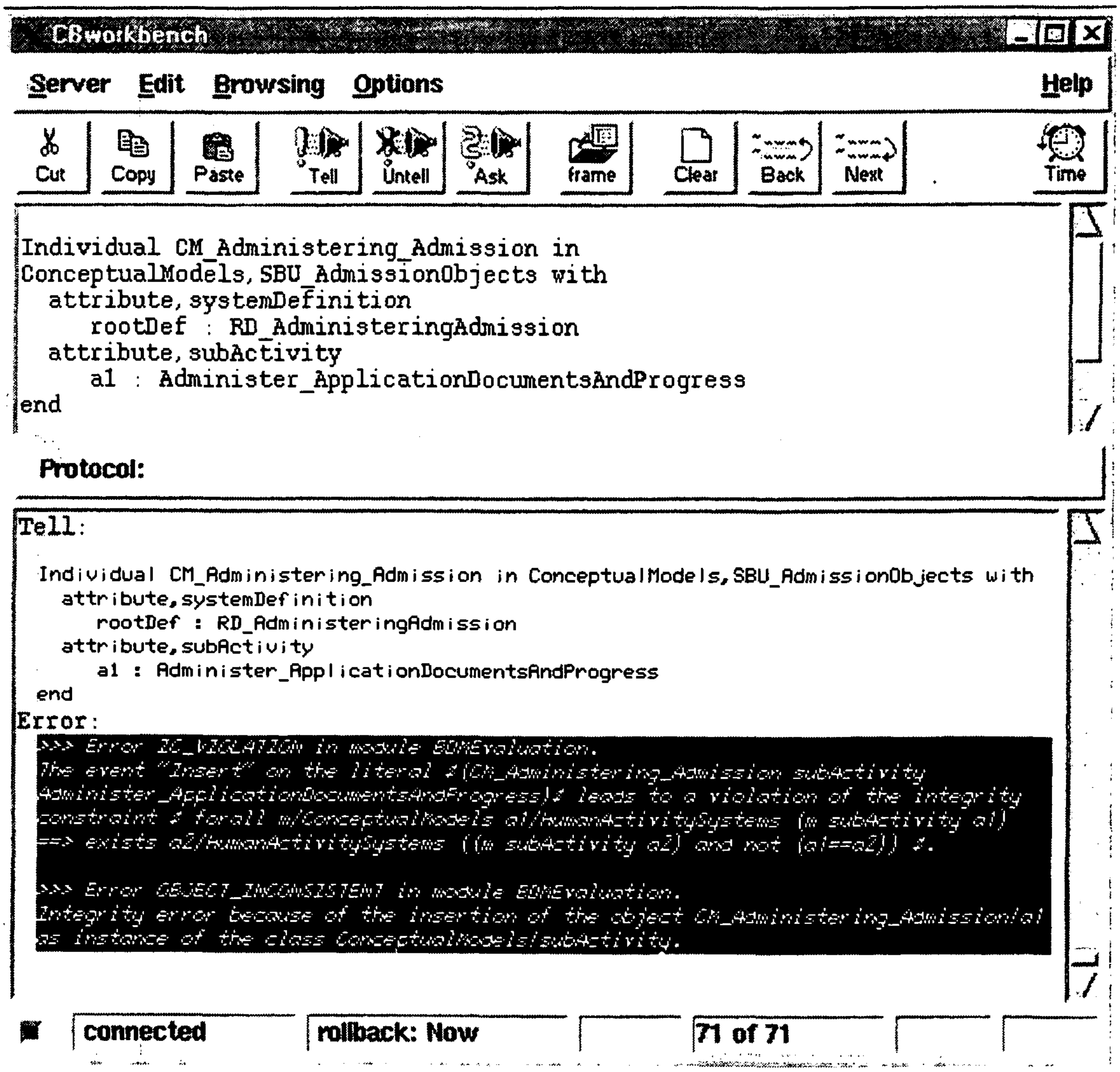

Figure 4: An object violating specified rules/constraint is rejected by ConceptBase

BENEFITS OF USING THE META MODEL

The goal of building the meta model is to use it as a unifying modelling language to model a problem domain, which is abstracted with a combined set of SSM and information models' constructs. Having said that, the proposed approach (viz. the unifying modelling language and the supporting software tool) is useful for documenting the post fact outcomes of the conceptualising activity, including the process of SSM. Specifically, a model that has been represented using the language and stored in a ConceptBase object base, documents only a particular objectively held view of a problem situation.

Continuing the SCORE model example, the use and benefits of the proposed approach can be shown. In a case study, the SCORE meta model is utilised to model the problem domain of the admission of students for university courses. For more details of this case study see Dang (1997). The Telos objects that represent the SCORE meta model and the resultant model of the university admission can be seen as a hierarchy (see Figure 5).

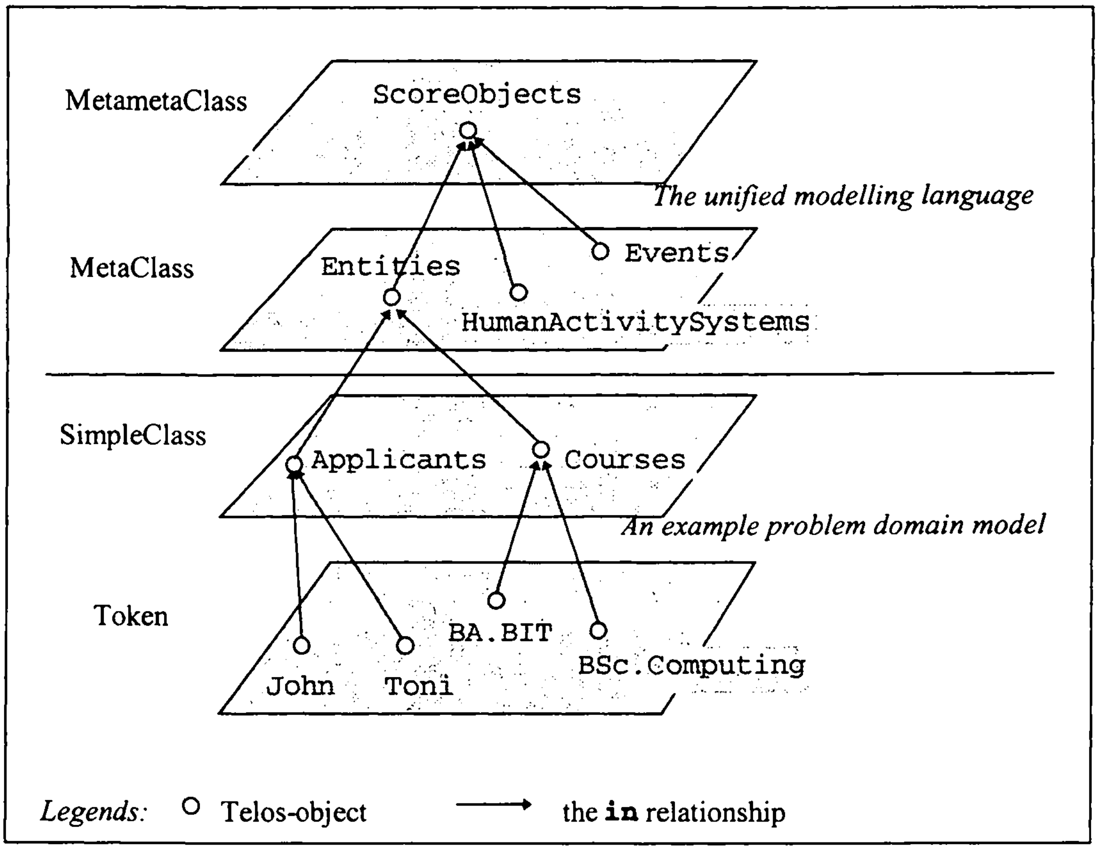

Figure 5: The hierarchy of the objects representing the SCORE model

Figure 5 illustrates the Telos feature that there is no limitation of the classification/instantiation hierarchy, i.e. classes can belong to other (*meta-*) classes which can also belong to other (*metameta-*) classes. It is this feature that imposes the problem domain model to inherit the semantic and structural properties of the modelling language defined by the meta model.

To build a model of a problem domain, the objects of the meta model (e.g. HumanActivitySystems, Entities), which are metaclasses, are instantiated with objects to represent the problem domain's concepts (e.g. Applicants and Courses). The resultant objects are simpleclasses that contain only tokens as their instances. Tokens are objects that have no instances. Tokens are employed to represent concrete objects of a problem domain, e.g. the applicant John and the course B.Sc. Computing, which are abstracted using the problem domain's concepts (viz. Applicants and Courses in this example, see Figure 5). Altogether, the SCORE meta model (i.e. the unified modelling language), the problem domain model (i.e. the abstraction of problem domains in terms of concepts) and the concrete data of problem domains are uniformly represented as Telos objects.

The uniform and formalised representation of the meta model and the problem domain model as Telos objects, which can be stored in the ConceptBase system, enables the modeller to manage the knowledge captured by the model. A model's objects can be retrieved not only in the textual format of Telos object specifications but also in graphical format. For example, the object ConceptualModels specified in Figure 1 can be visualised as shown in Figure 6.

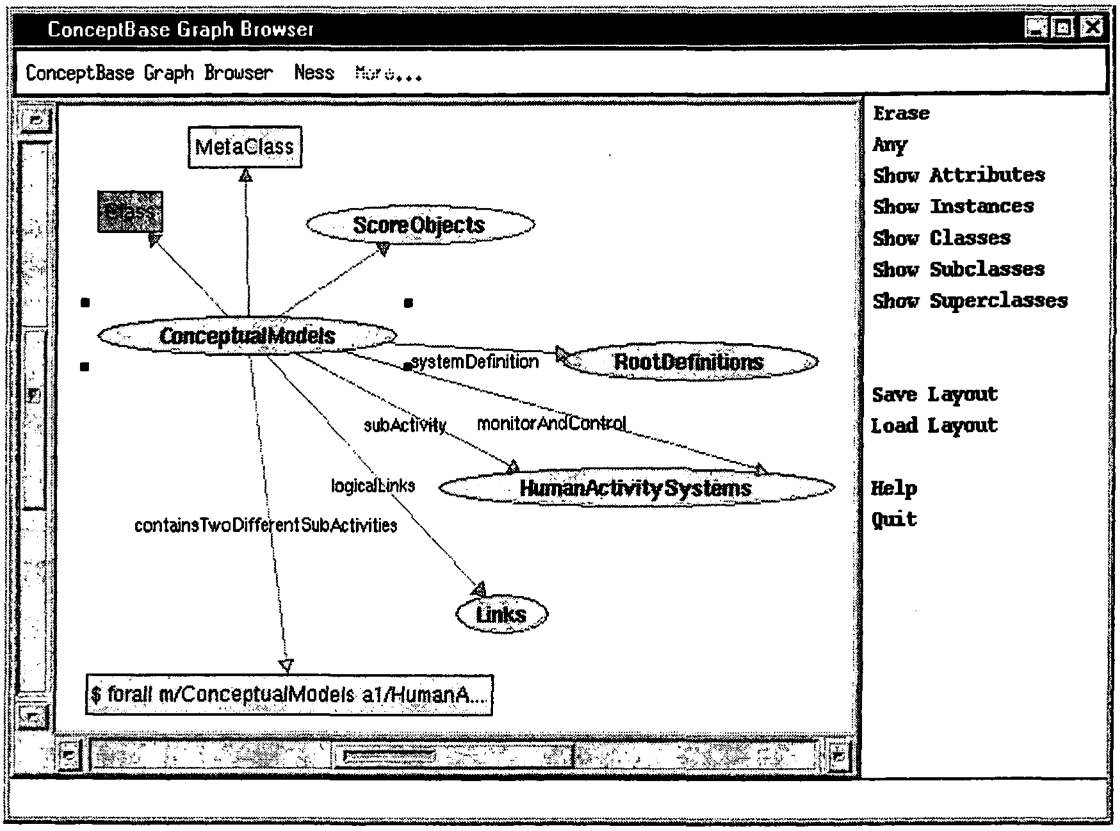

Figure 6: Visualising an object with its classes and attributes

Using the described method, the linkage between SSM's and information models becomes seamless because their links have been incorporated in the meta model, whose objects are subsequently instantiated to model problem domains. The example in Figure 7 shows that different aspects and relationships of the admission work are explicitly captured with Telos object specifications. In the example, one can see not only individual modelled objects, such as the human activity Handle_applicants_enquiries, the entities Applicants, SBU_CentralAdmissionOffice and the event ReceiveEnquiries, but also the links between them expressed by the attributes e.g. underlyingActivity, involvedEntity and involvedEvent.

```
Individual Applicants in
SimpleClass,Entities,SBU_AdmissionObjects,Class
  attribute
     who : "People who apply for courses";
     surname : String
end

Individual ReceiveEnquiries in
SimpleClass,Events,SBU_AdmissionObjects,Class with
  attribute,underlyingActivity
     a1 : Handle_applicants_enquiries
end

Individual Handle_applicants_enquiries in
HumanActivitySystems,SimpleClass,SBU_AdmissionObjects,Class
  attribute,involvedEntity
     iEty1 : Applicants;
     iEty2 : SBU_CentralAdmissionOffice
  attribute,involvedEvent
     iEvt1 : ReceiveEnquiries;
     iEvt2 : RecordAnEnquiries
end
```

Figure 7: Unification of modelled aspects at the representation level

The explicit capture of links in the model is seen by the fact that questions (about the knowledge captured by the model) like the ones listed below can be answered, retrieving data from the problem domain's model object base.

- Which entities share a given event?
- Which human activity systems share a given event?
- Which events happen in the performance of a given human activity system?
- Which entities are involved in the performance of a given human activity system?
- In which human activity systems does a given entity participate?
- Which rules and constraints govern the performance of a given human activity system?

Answers to these questions can be automatically obtained using Telos query classes and the *Display Queries* tool of ConceptBase. For example, Figure 8 illustrates the use of a query to retrieve the events, which are modelled and stored in the ConceptBase knowledge base, involved in the human activity Handle_applicants_enquiries.

SUMMARY AND DISCUSSIONS

In this paper we have proposed an approach to achieve a uniform and formalised representation of a model that combines SSM's models and information systems models. The approach is motivated by the idea that the linking of SSM's and information models can be enhanced by the explicit capture of the links by a final model that has a uniform formalised representation. The idea is implemented using meta modelling techniques and the conceptual modelling language Telos.

In comparison to other approaches that attempt to link soft systems and information modelling, our approach has several advantageous features. First, it is capable of uniformly representing the knowledge gained from the soft systems analysis as well as the knowledge captured by information modelling concepts. Second, the approach does not reduce SSM into a reductionist method, but it supports the soft systems model's extendibility and SSM incremental learning. Finally, the representation of the unified model is formal in the form of Telos objects, resulting in a model that can be maintained and exploited using a knowledge base management system (viz. ConceptBase). Having these features, the proposed approach has thus overcome the drawbacks highlighted in the second section of this paper.

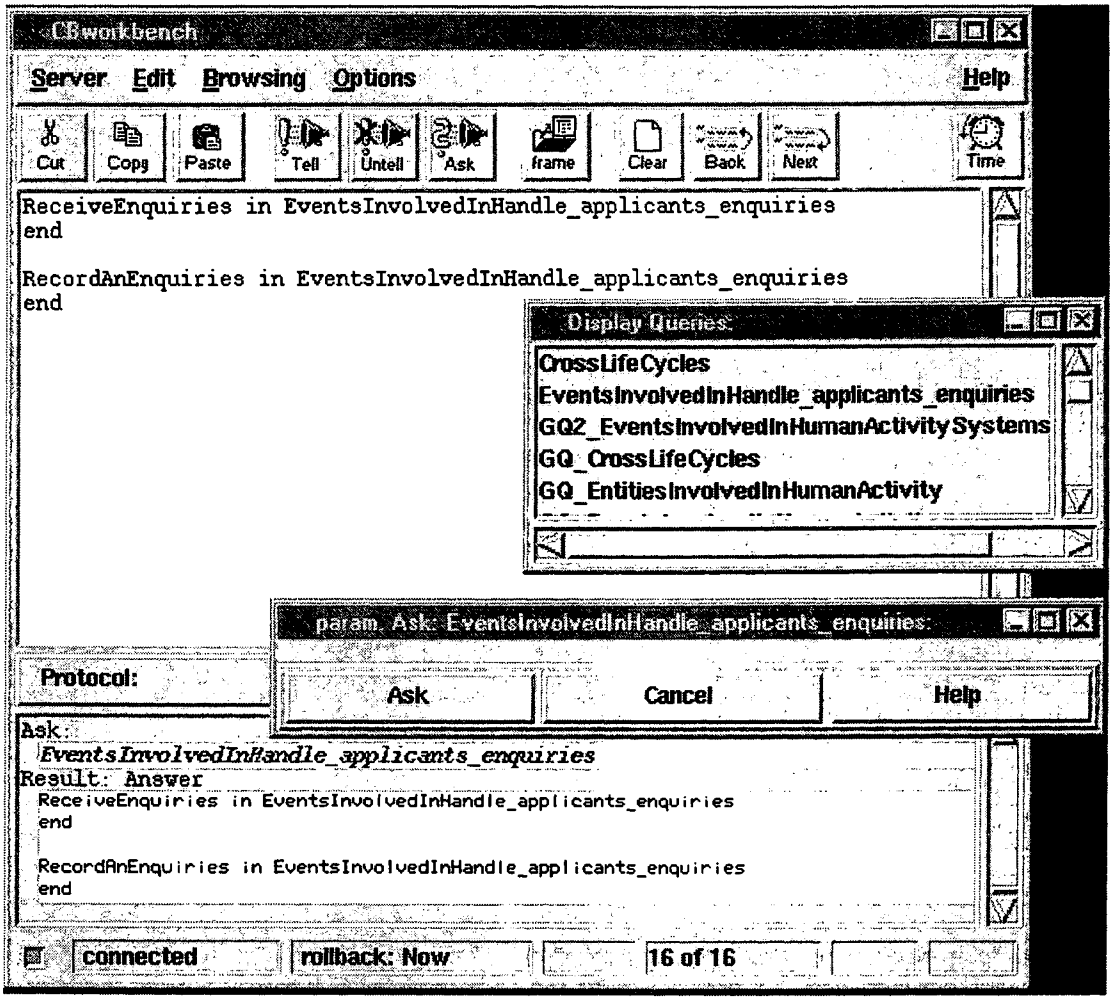

Figure 8: An example of a query class and its answered objects

Referring to the '*grafting*' and '*embedding*' taxonomy (Miles, 1988), the meta modelling approach belongs to neither of the classes of approaches. This is because our approach does not address the issues of linking SSM models with an information modelling approach or design method *at the conceptual level*. The meta modelling approach deals only with the *representation* of models that combine SSM and information models. It takes a combined model as given and provides a method to produce a unifying modelling language derived from the given combined model. Thus the method to generate a unifying modelling language of a combined model is applicable to models of both 'grafting' and 'embedding' approaches.

In terms of the links between SSM and information models, the method outlined in the paper deals with the representation of only those links between SSM's and information models that have been purposively established by the modeller at the conceptual level. As such, the meta modelling method proposed by this approach may be utilised as a complementary component to achieve a uniform formalised representation of models combining SSM's and information models at the conceptual level proposed by other approaches such as Multiview and Savage & Mingers' (1996).

Finally, because the use of the unifying language of a combined model requires the knowledge about Telos and the supporting software tool, it might be difficult for the modeller to use the language to build initial models of problem situations without using the SSM conventional modelling language. However, the unifying language can be used to create a model's objects repository. The model's objects stored in the repository can subsequently be exploited using the ConceptBase's tools for the analysis and reference purposes.

ACKNOWLEDGEMENTS

The author wishes to gratefully acknowledge the helpful discussions and contributory suggestions of Dilip Patel, Shushma Patel and Malcolm Crowe to improve this paper.

REFERENCES

Attardi, G. and Simi, M. (1981) Consistency and Completeness of Omega, a Logic for knowledge representation. **Proceedings of the 7th IJCAI**, Vancouver, 504-510.

Avison, D.E. and Wood-Harper, A.T. (1990) **Multiview: an exploration in information systems development**. Blackwell, Oxford.

Avison, D.E. and Fitzgerald, G. (1995) **Information Systems Development: Methodologies, Techniques and Tools**. 2nd edition. McGrawHill, London.

Brachman, R.J., Borgida, A., McGuinness, D.L., Patel-Schneider, P.F. and Resnick, L.A. (1992) The CLASSIC Knowledge Representation System, or, KL-ONE: The next generation. **Proceedings of the International Conference on Fifth Generation Systems**, Tokyo.

Brodie, M.L., Mylopoulos, J. and Schmidt, J.W. (Eds) (1984) **On conceptual modelling: Perspectives from Artificial Intelligence, Databases, and Programming Languages**. Springer-Verlag, New York.

CCTA (Central Computer and Telecommunication Agency) (1991) **SSADM Version 4 Reference Manual: Feasibility - Soft Systems Methodology**. CCTA, Norwich.

Checkland, P.B. and Griffin, R. (1970) Management information systems: a systems view. **Journal of Systems Engineering**, 1, 29-42.

Checkland, P.B. (1981) **Systems thinking, systems practice**. John Wiley & Sons, Chichester.

Checkland, P.B. (1992) SSM in information systems design: History and some current issues. **Systemist**, 14, 90-92.

Checkland, P.B. (1995) Soft Systems Methodology and its relevance to the development of information systems. In **Stowell (1995)**.

Checkland, P.B. and Howell, S. (1998) **Information, systems and information systems: Making sense of the field**. Wiley, Chichester.

Dang, Q.C. (1997) **A soft-systems-conceived model with knowledge representation for information systems in the office environment**. Ph.D. thesis. South Bank University, London.

Flynn, D.J. and Diaz, O.F. (1996) **Information Modelling, an international perspective**. Prentice-Hall, London.

Galliers, R. (1992) Soft systems, scenarios, and the planning and development of information systems. **Systemist**, 14, 146-159.

Gregory, F. and Merali, Y. (1993) Inductions, modality and conceptual modelling. **Warwick Business School Research Bureau Research Paper 79**. Warwick University.

Jackson, M.A. (1983) **Systems Development**. Prentice-Hall, Hemel Hempstead.

Jarke, M. (Ed.) (1995) **ConceptBase V4.1 User Manual**. RWTH Aachen, Aachen.

Jayaratna, N. (1994) **Understanding and Evaluating Methodologies, NIMSAD: A Systemic Framework**. McGraw-Hill, Maidenhead.

Jeusfeld, M.A., Jarke, M., Nissen, H.W. and Staudt, M. (1997) ConceptBase: Managing Conceptual Models about Information Systems. In **Handbook on Architectures of Information Systems**, Bernus, P., Mertins, K. and Schmidt, G. (eds). Springer-Verlag, Berlin (in press).

Avison, D.E. and Fitzgerald, G. (Eds) (1993) **Journal of Information Systems, Volume 3, Issue 3**. Blackwell, Oxford.

Kottemann, J.E. and Konsynski, B.R. (1984) Dynamic metasystems for information systems development. **Proceedings of the 5th International Conference on Information Systems**, Arizona.

Loucopoulos, P. and Zicari, R. (Eds) (1992) **Conceptual modeling, Databases, and CASE: an integrated view of information systems development**. Wiley, New York.

Miles, R.K. (1988) Combining "soft" and "hard" systems practice: grafting or embedding?, **Journal of Applied Systems Analysis**, 15, 55-60.

Mingers, J. (1995) Using Soft Systems Methodology in the design of Information Systems. In **Stowell (1995)**, pp.18-46.

Mylopoulos, J., Borgida, A., Jarke, M. and Koubarakis, M. (1990) Telos: a language for representing knowledge about information systems. **ACM Transactions on Information Systems**, 8, 325-362.

Nilsson, N.J. (1982) **Principles of artificial intelligence**. Springer-Verlag, Berlin.

Prior, R. (1990) Deriving data flow diagrams from a "soft systems" conceptual model. **Systemist**, 12, 65-75.

Quinllian, M.R. (1968) Semantic memory. **Semantic Information Processing**, Minsky, M. (ed.). MIT Press, Cambridge, Mass..

Savage, A. and Mingers, J. (1996) A framework for linking Soft Systems Methodology (SSM) and Jackson System Development (JSD). **Information Systems Journal**, 6, 109-129.

Stowell, F. and West, D. (1994) **Client-led design, a systemic approach to information systems definition**. McGraw-hill, London.

Stowell, F. (Ed) (1995) **Information systems provision - the contribution of Soft Systems Methodology.** McGraw-hill, London.

Stowell, F. (Ed) (1992) **Systemist, Special Edition on Information Systems**, Volume 14, Issue 3.