Model Integration in Information Planning Tools

Alex A. Verrijn-Stuart & Guus J. Ramackers

Department of Computer Science
University of Leiden
The Netherlands

e-mail: verrynstuart@rulcri.LeidenUniv.nl

Abstract. Information Planning requires integration in two senses, (1) linking the information requirements identified in the business system into further application detail ['vertical integration'] and (2) comprehensive modelling of relevant aspects, such that consistency is maintained at all levels ['horizontal integration']. These requirements may be satisfied by a framework that recognizes both the organizational information usage and the formalized (computerizable) systems that will serve it. Metamodels are discussed that provide a formal basis for such an all-embracing approach. They are illustrated by a prototype tool for capturing an organizational description, from which a broad specification of application systems may be derived by semi-automated means.

Keywords: Information Planning, Information Systems Design, CASE-tools

1 INTRODUCTION

The term "Information System" (IS) is used in two senses. On the one hand, it may refer to the over-all organizational usage of information (including communication), on the other, it may concern a computerized application. In previous publications [13, 8, 7], we distinguished these under the names ***ISB*** ('*IS* in the *B*roader sense') and ***ISN*** ('*IS* in the *N*arrower sense'), respectively. The ISB covers all informational aspects of the 'Business System' (BS), irrespective of the availability of computerized support as such. An ISN may be the conceptual model or the specification of a computerized (sub)system. At the pragmatic level, the ISB consists of a number of interacting agents (person, departments, their interactions and so on), whereas an ISN is a computer program (package or module) allowing storage, updating, manipulation and retrieval of representations of information.

ISB and ISN belong to different organizational cultures and failure to distinguish between the two leads to difficulties in creating and maintaining of information resources and to under-utilization of the information resources that are available. This paper is concerned with the creation-and-maintenance problem.

The need to continuously monitor an organization's information requirements ('Information Planning') is generally recognized, although one rarely practices it as an on-going activity [9]. We shall present an integrated approach that not only will allow identification and formulation of such requirements, but actually may form the basis for linking the Information Plan to formal descriptions of potential or actual computer applications. This linking is referred to as 'vertical integration', as opposed to the 'horizontal integration' of views at one particular level of abstraction, say, interrelating the data model and functional specification in a conceptual IS description [10].

Note that vertical integration (also) should cover the expression of how the IP fits in the organization, in other words how a "technically" formulated ISN is "embedded" in the "organizationally" formulated ISB.

Building on our ISB\ISN Framework [8], the problem addressed here is best stated with reference to the following diagram.

ORGANIZATION : (1) BS ↔ (2) ISB ↔ (3) ISN

Information Planning (IP) implies information-oriented modelling of one's BS, such that the ISB is highlighted. Ideally, that model allows various (potential or existing) ISN to be identified in the subsequent "Information *Systems* Planning" (ISP) activity.

Step (1)→(2), in fact, is no more than a shading of physical reality, but does not per se require a statement of what ISN might be desired. However, in this stage, one normally does question the BS's information requirements in general. The possiblity of adapting one's way of working (e.g. by a reorganization) may well come up. Borrowing Lundeberg's terminology [5], we call the totality of such *information-affecting organizational modifications* "Change Analysis I".

Step (2)→(3), on the other hand, means that one considers what computerized systems one should install (or modify, if already existing). In other words, here one addresses the potentials of new forms of support. Hence, we refer to these *computer-related modifications* as "Change Analysis II".

In the next section we shall present metamodels that may serve as a formal basis for IP tools. These would be capable of interfacing with CASE tools for detailed analysis and design of any ISN to be embedded in the ISB.

2 INFORMATION PLANNING METAMODELS

An IP tool - in our view - must possess the following qualities:

1 Convenient interfacing for the information planning staff[1], such that all concepts and relationships are expressed in "organizational" terminology;
2 The underlying meta-model is rooted in the organizational semantics, but, at the same time, formalized on the basis of a well-founded, coherent theory.

---

[1] An organizational setup is assumed where IP is practised in a staff function reporting to the CEO, working jointly with divisional and/or departmental staff responsible for IS development and maintenance, although a different allocation of responsibilities might apply.

Formalization - apart from scientific soundness and elegance - provides means for consistency enforcement and completeness checking in a practical way. It also should enable further analysis of a model described when using the tool, e.g. in simulations or performance studies. Finally, a formalization-based IP tool should be capable of extension into tools for designing and (ideally) constructing/modifying the application systems (in fact, ISNs in our definition).

As an initial meta-model, we consider an organization as consisting of departments (named *Organization units*, or "OrgUnits"), who have organizational *Tasks*, for which they may employ *Resource Units* ("ResUnits", active resources: either persons or devices, and passive resources: either material of informational), by involvement in *Actions*; these Actions form part of the aforementioned Tasks (see Figure 1). Note that further sub-characterization of ResUnits and Actions may be considered so as to reflect real world roles (e.g. device: machine or computer; primary actions: production, investment, maintenance, and secondary actions: office, computation, information, etc.). Such detailing is important in applications, but need be mentioned here only in passing [9].

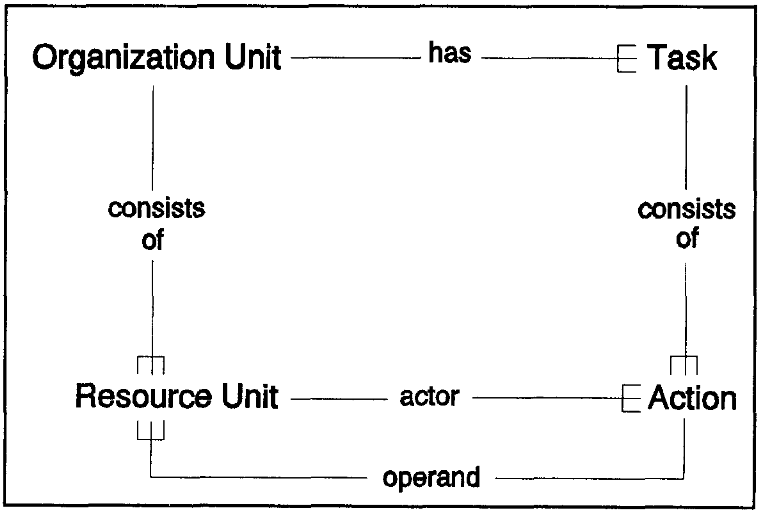

Figure 1: Initial Information Planning Meta-model.

To demonstrate the usefulness of these meta-models, a prototype IP tool, called **UNIS** (**UN**iversal **I**nformation **S**ystem description tool) was built [14]. It allowed a two-phase homing in on one or more ISN, by the following steps:

- *Organizational Description*: If the system description is empty, the user is able to populate it with OrgUnits, Tasks, ResUnits and Actions - if it is not empty, the population may be modified and extended.
- *Change Analysis I*: A given description may be subjected to a 'projection', by which any non-informational element (ResUnit, Task and/or Action) is pushed into the background; while engaged in this (otherwise automated) action, the user is asked whether fully physical elements (e.g. material streams or processes) need to be reflec-

The **TransCorp** case [14] concerns a haulage company, whose primary business is transporting parcel and bulk goods; among other things modelling is required of a temporary "Freight Storage" (the storage facilities). In one sample of the BS description this led to:

Task	*	[task-id 7]	Freight Storage	(a "primary" action in the real world)
Actions		[act-id 44]	Store Freight	(a "primary" action in the real world)
		[act-id 45]	Unstore Freight	(a "primary" action in the real world)
		[act-id 46]	Clean Store	(a "primary" action in the real world)
	*	[act-id 47]	Fill-in Storage Form	(a "support" action in the real world)
	*	[act-id 48]	Send Storage Form	(a "support" action in the real world)
		[act-id 51]	'Dummy/Empty'	(a [sofar] "undefined" action)

These actions are sequenced ('life cycle') : (46+51);44;47;48;45;47;48
where "+" stands for alternative action ("or"), and ";" for chaining

ResUnits	*	[unit-id 15]	Depot worker	(actor for act-id 44)
		[unit-id 20]	Truck	(operand for act-id 44)
		[unit-id 29]	Freight	(operand for act-id 44)
		....	....	....

No ResUnits had been associated with act-id 47-48; since the ISB highlights the "support" actions, while pushing into the background the real world activity, Change Analysis I might result in making the Storage Form handling explicit, as follows:

*	[unit-id 15]	Depot worker	(actor for act-id 47)
*	[unit-id 17]	Storage Form	(operand for act-id 47)
*	[unit-id 15]	Depot worker	(actor for act-id 48)
*	[unit-id 17]	Storage Form	(operand for act-id 48)

The items marked with an asterisk "*" would then appear in the *ISB* foreground.

Figure 2: Small portion of an ISB derivation and description.

ted in informational description terms (e.g. size of a stock, activity level of process, etc.); by reacting to these prompts, the user actially performs what amounts to "Change Analysis I", viz. introducing information flows and usage events, corresponding to decisions of new ways of (informational) working; the result is a description of the ISB, in which the various Actions may be chained as sequences and/or in parallel (see Figure 2).

- *Change Analysis II*: Given a "projected" system description of the ISB (i.e. where the informational elements have been indicated as such), the user is prompted to indicate whether any informational Actions should be computer-supported; deciding to do so constitutes Change Analysis II; if necessary, an informational device (computer) is "created"; the result is aan ISN description (see Figure 3).

The prototype (UNIS vs 1.0) was tested on a number of simple, yet non-trivial cases. This study revealed that the meta-model was quite satisfactory, but showed shortcomings in two respects. Firstly, descriptions took insufficient cognizance of data modelling and process modelling aspects (in fact, of the data, process and behaviour "perspectives", as a whole). These problems were referred to before under the generic name 'horizontal integration'. Secondly, the role of the ISNs was expressed such that each actor's involve-

In [14] a model had been given for the **TransCorp** case up to and including Change Analysis II. Whilst the explicit form of the resulting computerization is not relevant to this discussion, some of the choices are. They were as follows.

BS description:

Task	3	Global Planning	(a "support" task in the real world)
Action	17	Receive Order	(a "support" action in the real world)
	18	Cancel Order	(a "support" action in the real world)
	19	Confirm Order	(a "support" action in the real world)
	20	Send Order [adm]	(a "support" action in the real world)
	21	Send Cancel [client]	(a "support" action in the real world)
	36	Order Entry + Planning	(a "support" action in the real world)
	57	Receive Order [planner]	(a "support" action in the real world)

Action Sequencing (17|57);(18;(21|20))+(19;36)

["|" = independent action (concurrency), ";" = chaining, "+" = alternatives ("or")]

ResUnit	4	Order	("informational" operand: input)
	5	Planner	(actor: person)
	8	Global Plan	("informational" operand: output)
	10	Global Maintenance Plan	("informational" operand: output)

All actions and resunits in this example are associated with "support", hence are "informational"; when prompted to consider computer support (Change Analysis II), the choice was made to do so on behalf of the Planner, for action 36 (Order Entry for Planning, which leads to updated Global and Global Maintenance Plans, respectively).

Since, initially, the *ISB* did not have a computer facility embedded for use by the Planner (ResUnit 5), it is now introduced (**ResUnit 32: "I-ACTOR for ResUnit 5"**); upon selecting which operands are subjected to computerized support, the answer (in the treament of the case) is that the Order (ResUnit 4) is to serve as input and the two plans (ResUnits 8 and 10) will be output. Since (in the case) the latter two are only involved as output in connection with Entry (manual Action 36), Change Analysis II here leads to replacement of the original manual action by a computerized one (**New Action 58**).

Hence the **ISN description** will contain, among other things:

Task	3	Global Planning	(a "support" task in the real world)
Action	**58**	**Computerized Planning**	(resulting from **Change Analysis II**)
ResUnit	5	Planner	(User-Actor involved with **New Action 58**)
	32	**I-ACTOR for ResUnit 5**	(Computer made available for **New Action 58**)
	4	Order	(Input Operand for **New Action 58**)
	8	Global Plan	(Output Operand for **New Action 58**)
	10	Global Maintenance Plan	(Output Operand for **New Action 58**)

Figure 3: Small portion of an ISN derivation and description.

ment with computerized support would give rise to an independent Human-Computer Interaction situation ("HCI"); such a description is somewhat cumbersome, in that it implies that a large number of unrelated computerized systems might be constructed.

The data, process and behaviour models are, in fact, three different views on the same system. In the traditional design process, they are formulated separately, so that consistency enforcement becomes an external requirement. This is the aforementioned problem of 'horizontal integration'. Elsewhere, an object-based event-oriented model has been shown to provide a suitable approach for dealing with it [10]. We will now demonstrate how the meta-model may be expanded accordingly. Subsequently, the remaining 'vertical integration' problems (linking the BS, ISB and ISN descriptions) will be tackled.

The essence of the first meta-model was the way in which ResUnits may be "involved-in" Actions, viz. as Actors (one-to-many) or as Operands (many-to-one). That allowed formulating 'life-cycle' nets, describing such things as the handling of an order or the updating of a plan, i.e. *information flow*, as part of the Task description. Processing (in the sense of *data processing*) is implied in the concept of an "informational" Action. Whilst simple views of the behaviour models may be derived, the basis for the data model need to be extended, and some sophistication is required regarding the expression of information flows between OrgUnits, in order to specify the process model. The latter is achieved by introducing the concept Transaction (Figure 4).

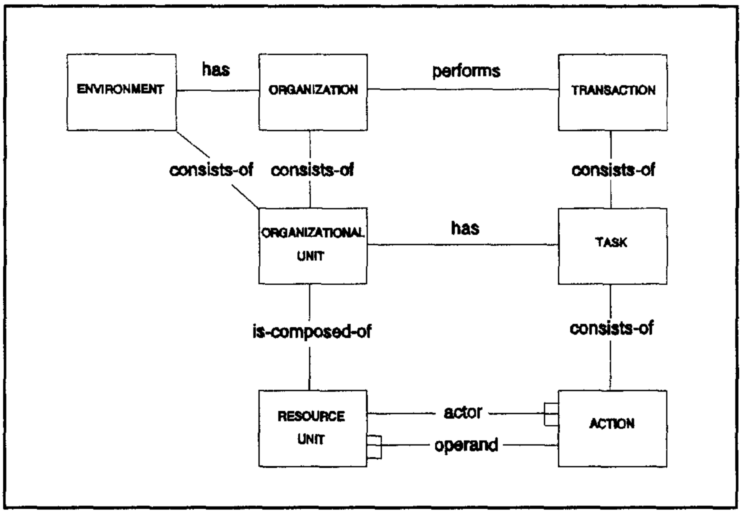

Figure 4: Meta-Model extension [1] (introducing Transactions).

The extensions required of the meta-model are, in fact, limited. They consist of making the "actor" and "operand" relationships of Figure 1 explicit and introducing suitable sub-typing. This modification is illustrated in Figure 5.

There is, however, a further need for extension, viz. an explicit meta-modelling of the 'life-cycle' involvement of Transactions, Tasks and Informational Objects. A nominal way of doing this is shown in Figure 6, which should be read as extending Figure 5 on the right hand side.

Finally, we need to make provisions for 'vertical integration'. As said before, that means (1) linking BS into ISB and (2) linking ISB into ISN. Requirement (1) has been met in the foregoing. Requirement (2) involves being more explicit as regards the role of computer support. This may be taken care of as follows.

Instead of a symmetrical position of Person-actors and Device-actors (both being "subject-of" some Action), one relates the Action directly, and asymmetrically, with the "primary"

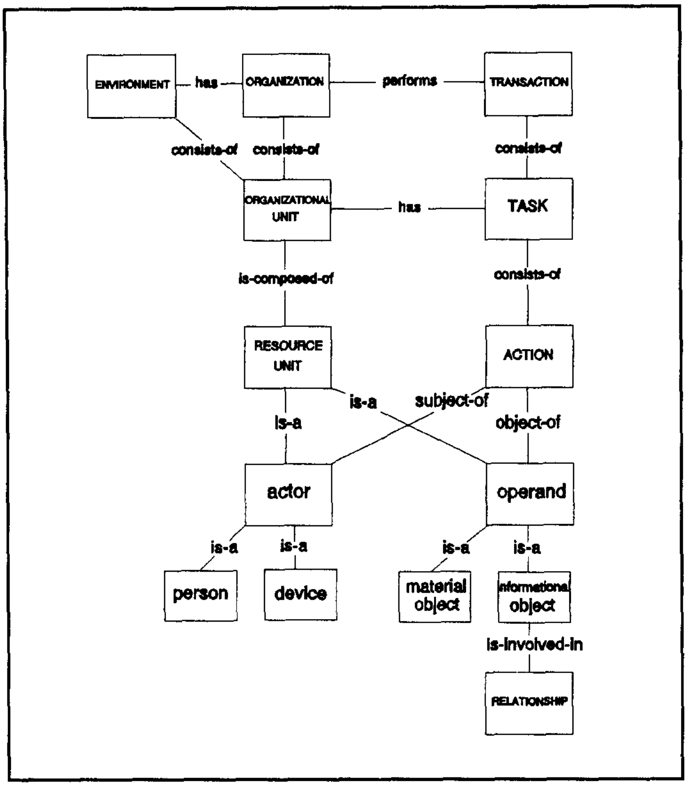

Figure 5: Meta-model extension -2- (adding data aspects)

Person-actor, on the one hand, and with the "co-actor" Device-informational (i.e. the computer), on the other. The latter is only necessary when an ISN is to be defined, but may be omitted when an implementation-free ISB is considered (see Figure 7).

Having formulated a BS in the foregoing terms and subsequently restricting onself to the informational aspects and specifying which actions require computerized support, i.e. expressing the model in the form of an ISB with embedded ISNs, the scene is set for more detailed system analysis. The ISN model described provides (1) an IS system architecture (consisting of initial analysis objects) and (2) a general organizational interaction context as input for systems analysis activity.

(1) A distributed *data model* is implied in the various allocated information operands. It is distributed in the sense that separate collections are associated with each OrgUnit, consisting of "local" entities, but possibly "shared" forms, with relationships defined for all. The ISN action *functionalities* - which may be defined in one's application as, e.g. "planning", "decision", "knowledge", "group", "transaction", etc. - can then be mapped to application objects. Again, these are distributed by OrgUnit.

Together, the sets of information and application objects constitute the *information system architecture*. Adding common objects, such as "archive" and "mailbox", and associating this structure with the hardware and generic software structure available, an initial model for full system analysis and subsequent design is specified.

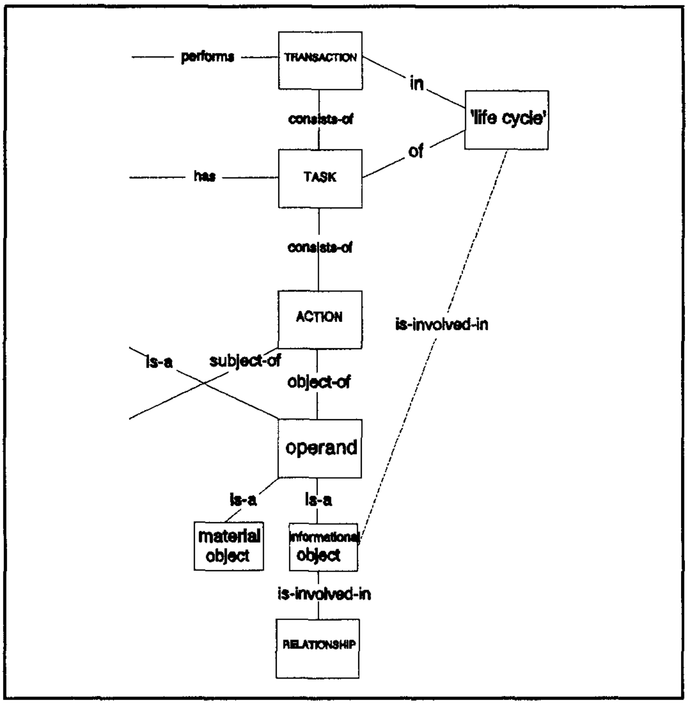

Figure 6: Meta-model extension -3- ('life cycle' added).

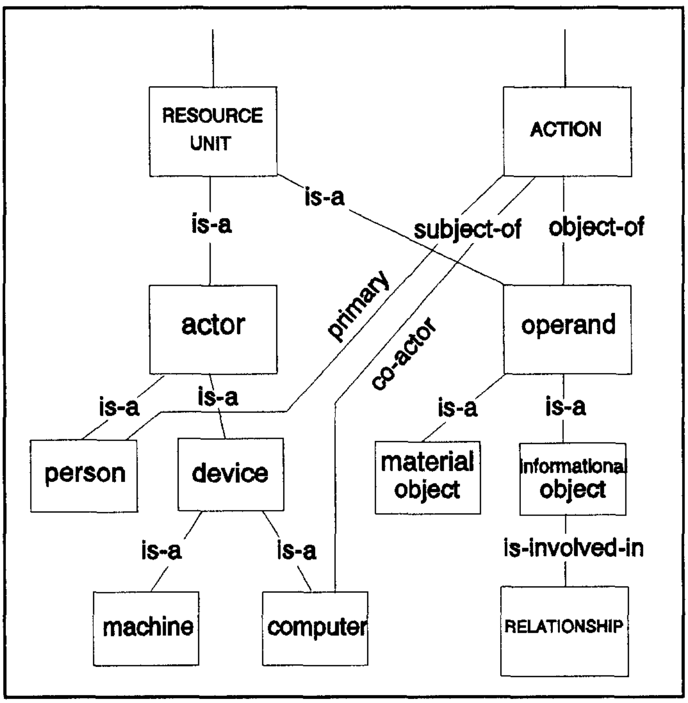

Figure 7: Meta-model extension -4- ("asymmetrical" ISN basis)

(2) Further analysis will lead to the specification of use-cases and office procedures, serving as descriptions of the interaction facilities for the user. The use-cases derive from the original Tasks and Transactions, and thus will be explicitly related to ISN actions, with their specified sequences. Likewise, the life cycle descriptions of information objects associate these with general use-cases, although their behaviour undoubtedly will have to be further refined. In this way, the Tasks, Transactions and Life Cycles relate the resulting system analysis model to the original Information Plan. This connection provides a basis for establishing specification control.

System analysis may be thus be performed on an "evolutionary" basis until all objects and interactions are specified that are required by the intended system users. If the specification is made in terms of an integrated description language (e.g. along the lines of [10]), a model is achieved that permits the final system design to be made.

3 PROTOTYPE EXPERIENCE

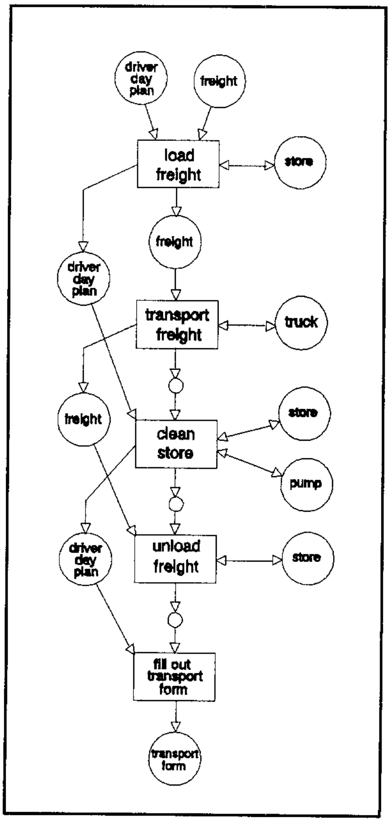

Figure 8a: BS Task

The graphical representations of the preceding section may be transformed into more specific definitions without much effort. For instance, in [9], various definitions were stated in BNF, allowing a variety of specific terminals that were useful for practical application. Such precise formal statements, in turn, facilitated programming the UNIS vs 1.0 prototype. Actual usage for *Organizational Description*, *Change Analysis I* and *Change Analysis II* turned out to be remarkably easy and rapid, in spite of the shortcomings discussed above. What was most missed - and what is intended to be introduced in the next version - is a more extensive graphical interface.

Results such as would have been achievable by the extended tool were worked out manually and would have been as illustrated in Figures 8 and 9. These show the BS and ISB views of a task (Fig. 8a-c) for the same test case as that from which the examples of Figures 2 and 3 were drawn, and an ISN task (i.e. the life cycle of an actor: Fig. 9). Figures 8 b-c show how (possibly automatic) "shading" portions of the Task diagram highlights the informational aspects. That demonstrates, for instance, a view of what Change-Analysis-I-options exist in a BS model or where an ISN my be introduced in Change Analysis II. Actors (humans or computers) are not shown, so as not to clutter up the diagram.

Our prototype study has demonstrated that integrated information planning is feasible, in principle. In particular, it is concluded that

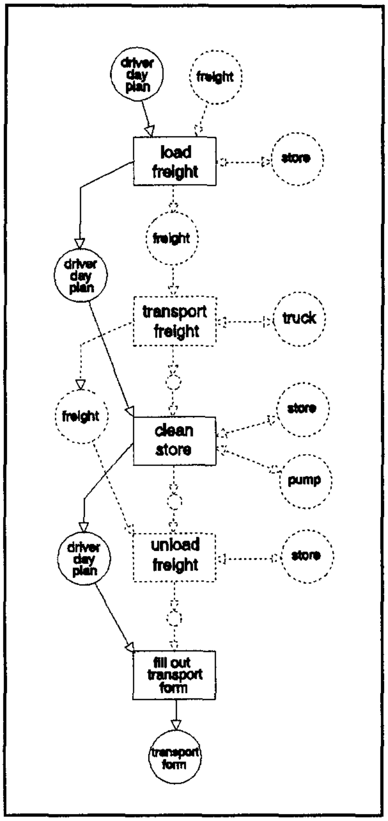

Figure 8b: ISB Task

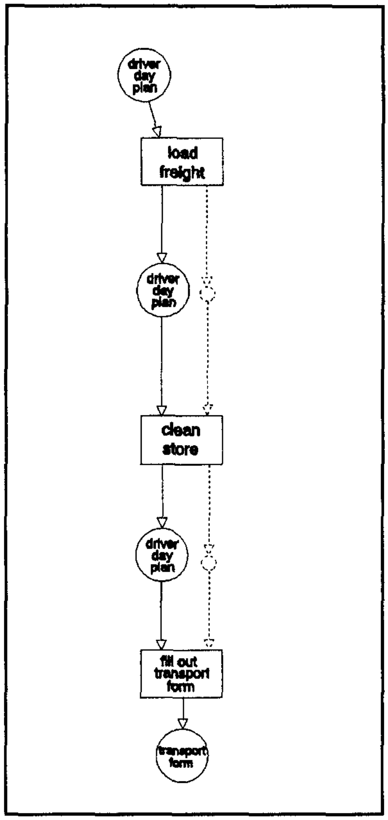

Figure 8c: ISB view

The models illustrated in Figures 8 a-c are part of the **TransCorp** case task "Garage: Transport Freight". The BS includes all physical elements. The ISB shows how the "Driver Day Plan" and "Freight" (to be transported) trigger subsequent activity, with physical aspects shaded into the background. If cleaning of the store is not required for a particular freight, that action is empty. The "pure" ISB view displays informational activity only, with physical element summarized.

1. *Organizational embedding* of the ISN may be strongly supported; based on the underlying integrated BS\ISB\ISN meta-model, both the ISB and ISN are describable in *integrated views* (BS\ISB and ISB\ISN, respectively);
2. *Evolution* of an overall model is strongly supported at both the ISB and ISN levels; modification of the BS\ISB view corresponds to *Change Analysis I* (logical business model), evolution of the ISB\ISN view by introducing or modifying the computer support activity constitutes *Change Analysis II*;
3. The integrated model is *more effective* in support of system development because (i) a semantically richer information planning may be formulated, in *user terms* (i.e. the organization and its information function are modelled in terms that are not influenced by database or processing aspects as such) and (ii) a richer interface is provided to systems analysis, consisting of a *system architecture* and a *system behaviour context*.

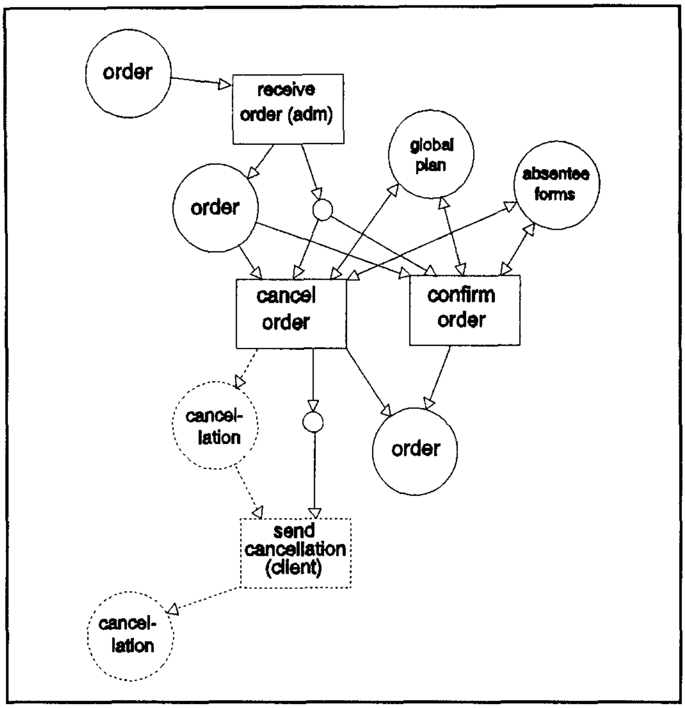

Figure 9: TransCorp Global Planning (ISN: full lines, non-computerized ISB: dotted lines)

Since the objects and relations in our meta-model remain related throughout, an IP tool as described conforms - in priciple - to the concept of "Hyper-CASE" [12, 2]. However, it is felt that the detailed system analysis and design required better be handled by separate CASE tools, that merely interface with the main IP tool [9]. In order to maintain consistency (and control), descriptions under the higher and lower tools should be "time-stamped" across the interface.

4 DISCUSSION

On the basis of the aforementioned experience, it seems highly feasible that a fully fledged Information Planning tool of the kind envisaged may be developed. The following points will then be considered.

The interface to (possibly automated) ISN system analysis (SA) tools will provide (1) an information architecture consisting of a distributed set of information objects and application objects (based on action functionalities), together with additional architecture objects (if any) and (2) a global context for adding system behaviour, applicable to use-cases to be associated with actions embedded in tasks and transactions, and forming a basis for further refinement of information object life cycles.

In order to be comprehensive, the SA tool should be "object-based" (e.g. as formulated in [10]). Then, it will be capable of adding "integrated" objects, that is to say, objects allowing each user point of view. Any such objects will be endowed with attributes, as required (e.g. order: client, freight, distribution date, etc.). Additionally, specialization and other relationships may be established. Finally, the dynamics will have to be introduced by describing 'use-cases' and 'life-cycles', involving the objects. These must be connected to actions at the ISN level.

A great advantage of an IP tool of this nature is that it allows a basic form of simulation of Place-Transition (P/T) nets (with multi-set places), since the task and transaction descriptions both use such nets. It should be noted that P/T nets [11] have less descriptive power than high level nets [4]. In particular, arc and transition inscriptions cannot be made, nor do tokens have explicit identity here. However, once these qualities are built into the formal model and the appropriate interface is installed, extensive animation studies may be envisaged (as e.g. in DESIGN/CPN [1] and ExSpect [3]). Thus, evolutionary analysis and continuous planning control will become possible that goes well beyond the ambitions of todays integrated tools (such as IEF [6]).

The manually worked extension of the UNIS vs 1.0 TransCorp case discussed and illustrated above, shows the kind of results that will be achieved. The detail of the specific modifications is under study and development of the next version will commence shortly.

References

1. K. Albrechts, K. Jensen and R. Shapiro, DESIGN/CPN, *A tool package suppporting the use of coloured Petri nets*, Meta Software Corp., Cambridge, Mass., U.S.A., 1988.
2. J.N. (Sjaak) Brinkkemper, *Formalisation of Information Systems Modelling*, Ph.D. Thesis, Catholic University, Nijmegen, Netherlands, 1990.
3. K.M. van Hee, L. Somers, M. Voorhoeve, *Executable Specifications for Distributed Information Systems*, in "Information Systems Concepts: An in-depth analysis", E.Falkenberg and P. Lindgreen, eds., North-Holland 1990.
4. K. Jensen, *Coloured Petri nets*, in "Petri Nets: Central models and their properties, W. Brauer, W. Reisig and G. Rozenberg, eds., LNCS nrs 254 & 255, Springer Verlag, 1986.
5. M. Lundeberg, G. Goldkuhl, A. Nilsson, *A Systematic Approach to Information Systems Development*, Inform. Systems **4** (1979) 1-12.
6. Ian G. Macdonald, *Automating the Information Engineering Methodology*™ *with the Information Engineering Facility*™, in "Computerized Assistance During the Information Systems Life Cycle", T.W. Olle, A.A. Verrijn-Stuart and L. Bhabuta, eds., North-Holland, 1988.
7. G.J. Ramackers & A.A. Verrijn-Stuart, *First and secord order dynamics in information systems*, Proceedings of the First International Working Conference on Dynamic Modelling of Information Systems, Noordwijkerhout, Netherlands, April 9-10, 1990 (also published by Elsevier Science Publishers, 1991).
8. G.J. Ramackers & A.A. Verrijn-Stuart, *The ISB-ISN framework of information systems* (Dept Comp.Sci., Univ. Leiden, Report 90-12).
9. G.J. Ramackers, K. Anzenhofer and A.A. Verrijn-Stuart, *Information Planning in the Office Environment* (Dept Comp.Sci., Univ. Leiden, Report 91-06).
10. G.J. Ramackers & A.A. Verrijn-Stuart, *Integrating information system perspectives with objects*, in "Object Oriented Approach in Information Systems", F. van Assche, B. Moulin and C. Rolland, eds., North-Holland 1991.
11. W. Reisig, *Petri Nets: An introduction*, EATCS Monographs on Theoretical Computer Science, Springer Verlag, 1985.
12. J.B. Smith, S.F. Weiss, *Hypertext*, Comm. ACM **31** (Nr 7, July 1988), 816-819 (Special issue on Hypertext).
13. A.A. Verrijn-Stuart, *The information system in the broader sense* (Dept Comp.Sci., Univ. Leiden, Report 89-13).
14. A.A. Verrijn-Stuart, UNIS: Prototype of a *UNiversal Information Systems description tool* (Dept Comp.Sci., Univ. Leiden, Report 91-05).