A Triptych Software Development Paradigm: Domain, Requirements and Software.

Towards a Model Development of A Decision Support System for Sustainable Development

Dines Bjørner

Department of Information Technology,
The Technical University of Denmark,
DK–2800 Lyngby, Denmark.

E–Mail: db@it.dtu.dk

Abstract. A paradigmatic three stage approach to software development is sketched in terms of a torso-like, but schematic development of informal and formal descriptions (i) of the **domain** of *sustainable development*, (ii) of **requirements** to *decision support software for developing models for and monitoring development* (claimed to be sustainable), and (iii) of rudiments of a **software architecture** for such a system.
In *"one bat we tackle three problems":* (i) illustrating a fundamental approach to separation of concerns in software development: From domain via requirements to software descriptions; (ii) contributing towards a theory of sustainable development: Bringing some precision to many terms fraught by "political correctness"; and (iii) providing, we believe, a proper way of relating geographic information system+demographic information system systems to decision support software. Perhaps a fourth result of this paper can be claimed: (iv) Showing, as we believe it does, the structural main parts of a proper presentation of software.

1 Introduction

A paradigmatic three stage approach to software development is sketched in terms of a torso-like, but schematic development of informal and formal descriptions (i) of the **domain** of *sustainable development*, (ii) of **requirements** to *decision support software for developing models for and monitoring development* (claimed to be sustainable), and (iii) of rudiments of a **software architecture** for such a system.

In *"one bat we tackle three problems":* (i) illustrating a fundamental approach to separation of concerns in software development: From domain via requirements to software descriptions; (ii) contributing towards a theory of sustainable development: Bringing some precision to many terms fraught by "political correctness"; and (iii) providing, we believe, a proper way of relating geographic information system+demographic information system systems to decision support software. Perhaps a fourth result of this paper can be claimed: (iv) Showing, as we believe it does, the structural main parts of a proper presentation of software.

The current paper primarily presents data models. They are in the style used in denotational and in algebraic semantics domain, respectively sort definitions. But we sketch some observer functions and some axioms. The notation used is that of RSL [1], the Raise Method's [2] Specification Language.

This paper is a torso: It sketches the application of a formal specification and refinement-based software development paradigm to a field either not very well understood or covered by AI (artificial intelligence) researchers and AI programmers. AI contributions, valuable as they may be, usually, as do most contributions in software engineering, zooms in on a narrow problem, solvable (ie. expressible, programmable) in some AI–language (or other). But usually such contributions do not try to isolate the domain from possible requirements; nor the requirements from the implementation. Instead the solution "drives" the development and its presentation.

We advocate a far more comprehensive approach. First we cover, in isolation, domain problems. In the jargon of software engineering these are the "up-stream" issues based on whose precise understanding one may formulate requirements. Then we relate domains to requirements, and finally to software (more precisely software architectures). Throughout we try to make more precise such software engineering and such international aid organisation jargon as decision support system, respectively sustainable development — as well as many other terms: indicator, equity, etc.

The triptych paradigm: from domains via requirements to software (descriptions, has been covered more comprehensively in other papers, for example the recent [3–9], and is claimed to have stood its first tests of usefulness by the well-reported work of the last seven years at UNU/IIST[1]. The present paper, to be fully appreciated by readers not familiar with formal development in the styles of VDM and RAISE, must therefore be complemented by the study of for example [10] or [2], preferably both. Then a professional reader can see how to turn the sketch of this paper into full reality.

[1] United Nations University International Institute for Software Technology, P.O.Box 3058, Macau: `http://www.unuiist.iist.unu.edu`

We leave it to others to compare the present approach to those of UML etc.

2 Summary Review

2.1 The Application

We *Domain analyse* (Section 4) the notions of Development as based on Resources. We then analyze the concept of Resources, their Attributes and Attribute Values and Indicators. Based on Value Indicators we define Equities. Sustainable Development is then defined in terms of Equities..

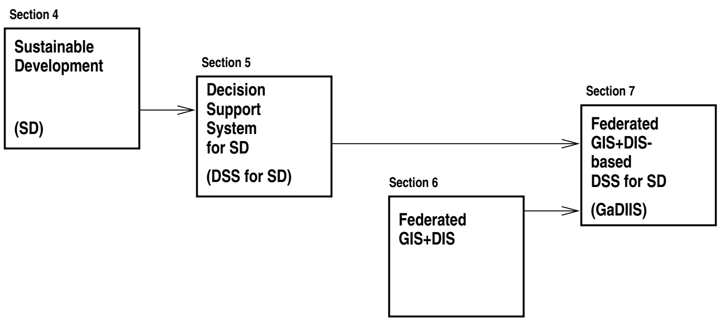
Based on these concepts we then analyze (Section 5) the Decision Making Processes and *Capture Requirements* for a Decision Support System for Sustainable Development (DSS for SD). In this section we introduce the notion of Resource Representations.

Independently we introduce (Section 6) a *Software Architecture* model for a Federated Geographic and spatially related Demographic Information System.

This model is then related (Section 7) to the DSS for SD system: $(\mathrm{GaD})^2\mathrm{I}^2\mathrm{S}$.

Section 4–7 thus relate (→) as shown in figure 1:

Fig. 1. Main Paper Section Relations



2.2 The Development Paradigm

It is here emphasized that the *Domain Analysis* of Section 4 does not refer to any software, nor to any computing or communications support.

It is strictly an analysis, and a formal model, of the concept of Sustainable Development and its constituent notions. Only in Section 5 do we refer, rather implicitly, to software, computing and communications support.

It is also to be emphasized that we do not refer to any conventional notion of geographic information systems or demographic information systems. Thus Section 4, perhaps rather surprisingly to many readers, does not assume geographic information systems or demographic information systems. That "connection" is only made in the last technical section, Section 7. To prepare for that, Section 6 "speaks" solely of geographic information systems and demographic information systems — with no reference to Section 4's or 5's decision support system for sustainable development!

This decomposition of the problem is a main contribution of this paper as are the models of Sections 4–7, in decreasing order!

3 Introduction

3.1 Background, Aims and Objectives

This paper has three objectives:

A Triptych Software Paradigm: We wish to illustrate the triptych notions of:

- *domain engineering*,
- *requirements engineering* and
- *software design*

Domain engineering builds a theory of the application domain. It does so by describing it: Informally and formally. As it, the domain, is, without any reference to computing, ie. also without any reference to requirements. Normally a domain is described normatively: encompassing many actual as well as possible instantiations. And the domain need be described from the point of view of all relevant stake-holders, and at a variety of abstractions: the very basics, the domain with its support technologies, with its rules & regulations, human behaviours, etc.

Requirements engineering builds a theory of some software for the support of some activities within the domain. It does so by describing domain requirements, interface requirements and machine requirements. *Domain requirements* projects and instantiates the normative domains;

and, in cases, also extends it. *Interface requirements* specify the human-computer interface (HCI): the way human users "see" the system (multimedia), dialogues between man and machine, etc. *Machine requirements* specify dependability (availability, accessibility, security, reliability, etc.), performance, and maintainability (perfective, adaptive and corrective), as well as development and execution platforms.

Finally *software design* specify the architecture of th software: How users and other software perceive or use it, its organisation: How the internal interfaces are composed. Architecture usually 'implements' domain and interface requirements. Program organisation 'implements' machine requirements.

Decision Support Systems: We attempt to combine two main streams of software technology: decision supports systems and geographic information systems in the specific context of environmentally sustainable development.

Sustainable Development: The main text of this paper will deal with this subject. The next sections will detail the points to be made.

3.2 Sustainable Development

The concept of sustainable development was brought into focus at the United Nations Conference on Environment and Development. That conference was held in Rio de Janeiro, Brasil, in June 1992.

An important document [11] submitted to that conference, and a document whose main enunciation, namely a definition of the concept of sustainable development, became a cornerstone of the result of the conference, was commonly known as the Brundtland Report.

The final document of the conference was the Agenda'21 report [12].

Definition 1 *Sustainable Development is development that meets the needs of the present without compromising the ability of future generations to meet their own needs. [11]*

It seems assumed in the above definition that it is indeed possible *to meet the needs of the present without compromising the ability of future generations to meet their own needs*!

From [13] we lift the quote taken from [14]:

Quotation 1 *Sustainable Development is a process of social and economic betterment that satisfies the needs and values of all interest groups,* *while maintaining future options and conserving natural resources and diversity.*

The above appears to have been a "first" definition of sustainable development. It also appears that it did not drew much attention. The next characterisation is due to [15]:

Characterisation 1 *Sustainable Development does not mean no development. It means improving methods for resource management in an environment of increasing demand for resource.*

It was referred to in [16]. The next quotation is due to [17]:

Characterisation 2 *Sustainability means that the evolution and development of the future should be based on continuing and renewable processes and not on the exploitation and exhaustion of the principal or the capital of living resource base.*

It was also referred to in [16]. The last characterisation is due to [18]:

Characterisation 3 *There are over 70 different definitions of sustainable development, offering a number of possible modifications of the development process and a number of different reasons for doing so.*

Also this was quoted in [16]. Finally we quote from:

`Source: Paul Samson, July 1995`
`http://greencross.unige.ch/greencross/digiforum/concept.html`

Quotation 2 *Sustainable development is currently a "catch-word"*[2]*, and as such, is often used and abused. Therefore, before we one can examine an issue of sustainable development, it is necessary to examine the concept itself. Some parameters for defining the concept are given here, and a number of competing visions are offered in the spirit of pluralism.*

The concept of, as opposed to the term of, "sustainable development" is not new; the profound and complex problems subsumed by the term can be traced back to the earliest human civilizations and the perennial tension between population growth and economic development, on the one hand, and the use of natural resources and ecosystems on the other. There is strong evidence suggesting that sustainable development constituted a challenge to our earliest societies, dating back to the ancient Sumerian,

[2] The use of double quote: "..." is Paul Samson's

Mayan and Mediterranean civilizations [19]. The term "sustainable development", however, is a recent invention, coming into common usage following the publication of the Brundtland Report [11], although even the term's origins may be traced back to before the 1972 United Nations Conference on the Human Environment [20]. The Brundtland Commission is also responsible for the most frequently cited definition of sustainable development: to meet the needs of the present without compromising the ability of future generations to meet their own needs. *As this section emphasizes, such a definition can be interpreted to have various meanings and is of little use if it is not placed within a specific context, or if the assumptions lying behind it are not clear. Indeed, as the following paragraphs will show a central point of this chapter is that the concept of sustainable development has multiple meanings, and that each is equally legitimate.*

It is noteworthy that a universally accepted definition does not exist for many basic concepts used by society, even for those which are seen to concern our well being. For example, it is often argued that the concept of security is useful precisely because its remains contested. This is why sustainable development, without a commonly accepted definition, appeals to virtually all groups who choose to participate in the environmental debate. Under such conditions, being "pro" sustainable development entails no risk or commitment to a specific set of goals or conditions since none are agreed upon [21]. Almost any group can find their own interest somewhere within the concept, and it is therefore hard to be against it in general. This allows the banner of sustainable development to be used by competing groups toward different or even contradictory ends. A number of these contradictions have been identified, and included among these are issues no less stark than "growth versus limits", "individual versus collective interests", "intergenerational versus intragenerational equity" and "adaptability versus resistance" [22]. However, these contradictions are part and parcel of human institutions and therefore, no less of Sustainability.

Further complication occurs because the concept of sustainable development can be broken into two parts. On the one hand, "Sustainability" relates to the question of the "carrying capacity" of the earth, while giving no attention to social issues, particularly those concerning equity and social justice. "Development", on the other hand, would appear to assume and even necessitate continual economic growth and ignore the question of ecological constraints or "carrying capacity". When these two concepts are put together, a very different one emerges, and the result is much more than the sum of the parts. It is therefore a multi-dimensional concept, and *it must be addressed at various levels simultaneously. Sustainability may be divide into three types: social, ecological and economic. The ecological definition is perhaps the clearest and most straightforward, measuring physical and biological processes and the continued functioning of ecosystems. Economic definitions are sharply contested between those who emphasize the "limits" to growth and carrying capacity, [23] and those who see essentially no limits [24].*

Similar to global environmental change, sustainable development remains first and foremost a social issue. Although the precise geo-spheric/-bio-spheric "limits" of the planet are unknown, it is suggested here that the limits to the globe's Sustainability for humans are more urgently social than they are physical. In other words, we will reach the social limits of Sustainability before we reach the physical ones. Thus, our focus should be on society-based solutions for managing the multiple aspects of global change rather than on technology-based ones. It is important to emphasize the human aspect of sustainable development — for example, institutional and political constraints.

Any conclusions about the meaning of sustainable development remain dependent on considerations of context and spatial and time delimitations. At a global level, the following set of definitions serves well:

> *In the narrowest sense, global Sustainability means indefinite survival of the human species across all the regions of the world... A broader sense of the meaning specifies that virtually all humans, once born, live to adulthood and that their lives have quality beyond mere biological survival... the broadest sense of global Sustainability includes the persistence of all components of the biosphere, even those with no apparent benefit to humanity [25].*

4 Sustainable Development — A Domain Analysis

We analyze the concept of sustainable development. The analysis is decomposed into a number of parts.

4.1 Development

Development is about resources: be they natural resources, monies people, equipment, capabilities, or other. "Raw" development is (like) a function: from a set of resources to a set of resources:

type

R
value
D': $R^* \xrightarrow{\sim} R^*$

In "raw" development we *just develop!* — without any consideration to resources at hand, in particular: *whether sustainable* or not! Two developments with exactly, if that was ever possible, resources need not yield the same resulting resources.

The above expresses that there is an abstract type, a sort, named R, which stands for all resources, and that there is some further unspecified, ie. "grossly" underspecified function (hence partial $\xrightarrow{\sim}$), D', from sequences of resources into sequences (*) of resources.

4.2 Resources

Resources "fall" in ("main") categories (C):

Examples 1 *Land, Monies, Minerals, Crops, People, etc.*

Each category (has a name and) designates a set of resources possessing same attributes (A):

Examples 2 *Land: quality, area, location, cost, . . . ; Monies: kind, currency, amount, . . . ; . . . ; People: profession, quality, quantity, . . . ; etc.*

Each category and attribute (pair) designates a value class (VAL):

Examples 3 *(l:Land,a:Area): from one acre to maybe 20,000 acres; (p:-People, a:Amount): from one to perhaps 2,000; etc.*

type
C, A, VAL
value
obs_RC: $R \to C$
obs_CRs: $C \to$ R-**set**
obs_RAs: $R \to$ A-**set**
obs_RAV: $R \times A \xrightarrow{\sim} VAL$
obs_AVs: $A \to$ VAL-**infset**
axiom
$\forall$ c:C • $\forall$ r,r$'$:R •
$\{r,r'\} \subseteq$ obs_CRs(c) $\Rightarrow$ obs_RAs(r) = obs_RAs(r$'$)
$\wedge$ obs_RC(r) = obs_RC(r$'$) = c $\wedge$ ...

The above RSL notation expresses that there are further algebraic sorts: categories, attributed and values, and that there are some further unspecified observer functions.

Each resource "belongs" to one main category (obs_RC).[3] Each resource relates to a set (**-set**) of attributes (obs_RAs). Any given resource may or may not have a value for a given attribute (obs_RAV).

Our (domain) observer functions are not definable by us, but are defined by the domain. Observer functions are, however, characterisable by two kinds of axioms. Firstly general axioms that characterise the general model of sustainable development. One is shown above: It expresses that all resources of a specific category must have the same set of attributes and, of course, be of that category. We may relax the former (same set of attributes), but cannot relax the latter (be of that category). (The symbols • and ⇒ can be "pronounced" 'such that' and 'implies/imply', respectively.) Secondly instantiated, specific axioms: For a given, ie. an instantiated case of sustainable development, for example the building of a chemical plant, or fertilisation of crops, the resources, categories, attributes and values are "fixed" and a (possibly) consistent, but not necessarily complete axiom scheme "set up": one which approximates relations that are believed to hold between these specific resources, categories, attributes and values.

4.3 Indicators

An indicator is a measure of desired values of resource attributes. Sometimes an indicator is a value with, perhaps, some fuzzy membership (or probability) function. Sometimes an indicator is a pair of values (a range) (perhaps adorned with some sort of fuzziness). And, sometimes an indicator is a function, for example a simple function from time to attribute values, or a more complex function, for example a function from time and another attribute value to (perhaps fuzzy) values.

An indicator thus expresses a desirable interval within which actual resource attribute values are to range at given times and/or in the presence of other (fuzzy) valued attributes, etc.

type
  I

[3] We could postulate another category-observer function (obs_RCs, not shown) which to resources associated "related" categories, such that, for example two different resources of the same main category associated to not necessarily the same set of "related" categories.

Fuzzy
value
is_in_Rng: R × A × I $\tilde{\rightarrow}$ Fuzzy

4.4 Resources, Attributes and Indicators

In development we are interested in certain resources, and for each of these, in certain attributes, and, for each of these, in focusing on certain (intervals of) indicators. We may speak of such a "thing" as a RAIs: A resource to attribute indicator range "table":

type
RAIs = R $\overrightarrow{m}$ (A $\overrightarrow{m}$ (I×I))

The above defines RAIs to be a space of maps from resources to maps from (their) attributes to pairs of ("lo–hi") indicators.

An 'abstract' example could be:

Examples 4 rais:

$$\left[\begin{array}{l} r_1 \mapsto \left[\begin{array}{l} a_{1_1} \mapsto (i_{1_{1_1}}, i_{1_{1_2}}) \\ a_{1_2} \mapsto (i_{1_{2_1}}, i_{1_{2_2}}) \end{array}\right] \\ r_2 \mapsto \left[a_{2_1} \mapsto (i_{2_{1_1}}, i_{2_{1_2}}) \right] \end{array}\right]$$

The example, rais:RAIs, expresses that we are concerned with exactly two resources, and, for resource r_1 in two of its attributes. We do not, in RAIs, express resource categories nor resource attribute values: these properties are part of the resources, r_1, respectively r_2. Cf. observation functions obs_RC, respectively obs_RAV, etc.

4.5 Equities: Constraints and Objectives

Sustainable Development: Development is said to be sustainable if (i) it maintains an invariant (an equity) between resources before and after development.

Other variants of what an equity is are: if (ii) it, after development, achieves certain (indicated) resource attribute values, respectively if (iii) development is constrained, 'before and after', by indicated attribute value ranges ("within interval").

An equity, E′, is therefore chosen to express a fuzzy (in general a multi-criteria, underspecified) relation.

type
  Small
  $E' = (RAIs \times RAIs) \xrightarrow{\sim} Fuzzy$
  $ES' = En \underset{m}{\rightarrow} E'$
  Acceptable = Fuzzy × Small → **Bool**

We do not mandate any specific equity relation. The construction of equity relations entail oftentimes rather serious mathematical, control-theoretic, operations-analytic, knowledge-based (expert) system, or other modeling (see Section 7.3).

When applying a fuzzy equity function to pairs of resource sets combined with their attributes and the indicators of these attributes: namely a pair designating a "before–after" (development) relation, we expect to get an acceptable level (below 'small'). Thus the class 'Acceptable' denotes predicates, each of which we supply with an acceptance factor ('small').

The primed type names, for example E' and ES', designate precursors for subsequent, stepwise "refined" unprimed type names. For E see Section 5.7.

4.6 Analysis = Modeling "in the Small"

Such modeling — as was just mentioned at the end of the previous section — may stabilize only after repeated analytical experiments.

That is: fixing which are the relevant indicators and which are the relevant equity functions require various kinds of mathematical modeling, i.e. analysis.

Analysis with respect to sustainable development involves:

1. identifying relevant resources (rs:RS),
2. affected attributes (a:A),
3. their indicator intervals ((li,hi):I×I),
4. specimen (analysis labeled (lbl:Lbl)) combinations of resources, attributes and indicators (rais:RAIs, and lrais:Lbl_RAIss)
5. relevant (named, En) equity functions (in ES′).

Formally, analysis amounts to:

type
  Lbl
  RS = R-**set**
  $Lbl\_RAIss = Lbl \underset{m}{\rightarrow} RAIs$

Analysis′ = RS × Lbl_RAIss × ES′
value
A_Result: Analysis′ $\xrightarrow{\sim}$ (En $\overrightarrow{m}$ Fuzzy)
axiom
[proper analysis]
∀ (rs,lraiss,es′):Analysis′ • ∀ rais:RAIs • rais ∈ **rng** lraiss
⇒ **dom** rais ⊆ rs ∧ ∀ e′:E′ • e′ ∈ **rng** es′ ⇒ (rais,) ∈ **dom** e′

The result of analysis associates with each equity some fuzzy judgment as to whether a planned development, as expressed by the equity functions, achieve equity. The keywords **dom** and **rng** designate the map definition (domain), respectively the range set yielding operations.

4.7 Planning

Planning is concerned with creating descriptions of development (functions, d:D). Since these have to satisfy a variety of equities, planning also involves analysis.

type
DS = Dn $\overrightarrow{m}$ D
D = RAIs $\xrightarrow{\sim}$ RAIs
Plan = Analysis′ × DS
axiom
∀ ((rs,nmrais,es′),ds):Plan • ∀ d:D, rais:RAIs • d ∈ **rng** ds
⇒ rais ∈ **dom** d ∧
let rais′ = d(rais) **in**
∀ e:E • e ∈ **rng** es′ ⇒ ∃ s:Small • Acceptable(e(rais,rais′),s) **end**

4.8 Sustainable Development

Sustainable development is now the act of actually carrying out the planned development after analysis of plans has validated these according to desired equities.

type
Development = (Plan × RS) $\xrightarrow{\sim}$ RS

Here we have taken a very simplistic view of development. A more realistic view would only add details and not further illustrate the formalisation principles we strive to adhere to.

4.9 Time Frames

Among the resources treated is time. In the RAIs arguments there will undoubtedly be various forms of time attributes and indicators: past, present and future time, time intervals, etc. Non-time attribute indicators may themselves be functions of times and intervals.

Thus we believe, that in the above model we capture "all" conceivable needs for time parameters, time considerations, etc.

4.10 Discussion

Resources vs. Resource Representations: As always our language of communication, in the daily pursuit of our business: here sustainable development, mixes references to "real" resources with references to representations of resources. As long as we are fully aware of the dangers in possibly confusing them, OK. So far we have been referring to "real" resources, not their representation. That will be done in Section 5.1.

Function Arguments and Results: In this paper we "lump" all conceivable arguments to functions and predicates into the convenient form of one single rais:RAIs argument.

Readers may find that when they start understanding what all these functions, like Equity "predicates", Experimental Analysis and Analysis functions, Planning and Development functions, are doing, then they may start wondering: *what happened to time considerations?; what happened to financial expenditures, what happened to the deployment of engineers, designers, construction workers, etc.?*

The simple answer is: They are all gathered together, not as separate parameters to conventionally type functions, as in mathematics or programming, but as a rais:RAIs argument.

Let us just show an example:

Examples 5 *A 'before'/'after' development relation:*

- *Before development:*

$$\begin{bmatrix} r_1 \mapsto [a \mapsto ii_a, t \mapsto ii_t] \\ r_2 \mapsto [a_\alpha \mapsto ii_\alpha] \end{bmatrix}$$

- *After development:*

$$\begin{bmatrix} r_1' \mapsto [a \mapsto ii_a', t \mapsto ii_t'] \\ r_2' \mapsto [a_\alpha \mapsto ii_\alpha] \\ r_3 \mapsto [a_\beta \mapsto ii_\beta] \end{bmatrix}$$

An interpretation of the above could be that in this development three resources are being changed (r_1, r_2) or created (r_3). Resource r_1 has a time attribute. Before development its value satisfied some equity indicator interval ii_t, afterwards is satisfies ii'_t. Etcetera. What we mean by 'satisfy' is again open for wide interpretation.

This example serves to show that the preparer, the analyzer and planner have very wide degree of freedom in formulating functions over almost any combination of resources and, within these, of interpretation.

5 Requirements Capture: A "DSS for SD"

By a 'DSS for SD' we mean a decision support system for sustainable development.

Section 4 described what we mean by sustainable development.

In this section we will analyze the actions needed in preparing for, making plans and analyzing plans for sustainable development, and, in particular, identify the computer and communications support of these actions. That is: we capture, in this section, the requirements for a DSS for SD.

In doing so we shall repeatedly refer to subsections of section 4.

5.1 Resource Representation

In section 4 we "dealt" with "real" resources. In real reality we "deal" with representations of resources. That is: we assume that every resource (r:R) that we wish to handle can be "formally" represented, ie. modelled by some rr:RR, the class of resource representations. We therefore redefine the functions over R to also apply to RR:

type
  RR
value
  obs_RRC: RR $\to$ C
  obs_RRAs: RR $\to$ A-**set**
  obs_RRAV: RR $\times$ A $\stackrel{\sim}{\to}$ VAL
  is_in_Rng: RR $\times$ A $\times$ I $\stackrel{\sim}{\to}$ Fuzzy

With this follows that we redefine:

type
  RRAIS = RR $\overrightarrow{m}$ (A $\overrightarrow{m}$ (I × I))
  E = (RRAIS × RRAIS) $\tilde{\rightarrow}$ Fuzzy

Etcetera.

5.2 Problem Synopsis

We refer to section 4.1.

The problem synopsis — in a "gross" way, to be detailed (detail-resolved) by subsequent actions — identifies (including names) the major (initially "raw") resources and development functions. **Text** stands for text that explains the pragmatics of whatever is being represented.

type
  Q /∗ text ∗/
  Resources = Q × (Rn $\overrightarrow{m}$ (Q × RR))
  DevtFct = (Q × (C^* × C^*))
  DevtFuncts = Q × (Dn $\overrightarrow{m}$ DevtFct)
  Synopsis = Q × Resources × DevtFuncts
  Location
value
  obs_RLoc: RR $\tilde{\rightarrow}$ Location

obs_RLoc is an observer function which to every resource representation associates its physical Location. Observe that only now did we actually use the notion of a resource category (c:C). When we, earlier, dealt with "real" resources there basically was no need to introduce categories of resources. Now that we work (mostly) with representations of resources, then we must introduce that type notion.

The overall problem synopsis is informally described (**Text**). Each resource and development function is named (Rn, Dn) and explained (**Text**), and, for the development functions, a "type"-definition of the function is given in terms of the resource categories involved. Resources themselves are, of course, not present in the decision support system for sustainable development "machinery": only representors (RR) which further locates the resources (etc.).

Requirements Capture 1 *Hence the decision support system for sustainable development must provide a repository (a data base) for 'Synopsis' as well as appropriate functions, for example for initializing PS, for* *inserting new, and for displaying, searching, sorting, updating, deleting existing resource representor and development function entries.*

5.3 Resource Mappings

We need establish mappings between real resources and their representors.

type

$\mathrm{RRRM}' = \mathrm{RR} \xrightarrow[m]{} \mathrm{R}$

$\mathrm{RRRM} = \{|\ \mathrm{rrrm} \mid \mathrm{rrrm}{:}\mathrm{RRRM}' \bullet \mathbf{dom}\ \mathrm{rrrm} = \mathrm{RRS}\ |\}$

$\mathrm{IRRRM}' = \mathrm{R} \xrightarrow[m]{} \mathrm{RR}\text{-}\mathbf{set}$

$\mathrm{IRRRM} = \{|\ \mathrm{irrrm} \mid \mathrm{irrrm}{:}\mathrm{IRRRM}' \bullet \cup\ \mathbf{rng}\ \mathrm{irrrm} \subseteq \mathrm{RRS}\ |\}$

$\mathrm{RMs}' = \mathrm{RRRM} \times \mathrm{IRRRM}$

$\mathrm{RMs} = \{|\ (\mathrm{rrrm},\mathrm{irrrm}) \mid$
$\quad (\mathrm{rrrm},\mathrm{irrrm}){:}\mathrm{RMs}' \bullet \forall\ \mathrm{rr}{:}\mathrm{RR} \bullet \mathrm{rr} \in \mathbf{dom}\ \mathrm{rrrm}$
$\quad \Rightarrow \mathrm{rrrm}(\mathrm{rr}) \in \mathbf{dom}\ \mathrm{irrrm} \wedge \mathrm{rrrm}(\mathrm{rr}) \in \mathrm{irrrm}(\mathrm{rrrm}(\mathrm{rr}))\ |\}$

The $\{|\ \mathsf{a} \mid \mathsf{a}{:}\mathsf{A} \bullet \mathsf{P}(\mathsf{a})\ |\}$ expression defines a sub-type of A, namely all those a of A that satisfy P(a). The prefix $\cup$ denotes distributed set union — since, in this case, **rng** irrrm yields a set of sets.

Requirements Capture 2 *Hence the decision support system for sustainable development must provide a repository (a data base) for these mapping as well as appropriate functions, for example for initializing RMs, for inserting new, and for displaying, searching, sorting, updating, deleting existing map entries.*

5.4 Resource Names and Resource Representations

Resources are clustered in categories and maps between representors of real resources and their (non-unique) denotations must be established:

type

$\mathrm{Resource\_Clusters} = \mathrm{C} \xrightarrow[m]{} \mathrm{Rn}\text{-}\mathbf{set}$

$\mathrm{RR\_R\_Mapping} = \mathrm{txt}{:}\mathrm{Q} \times \mathrm{RRRM}$

$\mathrm{R\_RR\_Relation} = \mathrm{txt}{:}\mathrm{Q} \times \mathrm{IRRRM}$

$\mathrm{Resource\_Info} = \mathrm{Resource\_Clusters} \times \mathrm{RR\_R\_Mapping} \times \mathrm{R\_RR\_Relation}$

Requirements Capture 3 *Hence the decision support system for decision support must provide a repository (a data base) for Resource_Info as well as appropriate functions, for example for initializing Resource_Info, for inserting new, and for displaying, searching, sorting, updating, deleting existing category and resource representor to "actual" resource mapping entries.*

5.5 Resource Attributes, Values and Indicators

We refer to sections 5.1 and 4.4.

For each resource category we must identify all relevant attributes, (Cluster_Atrs) and for each specific resource (identified by its name) and attribute the (Sustainability) indicators (Resource_Inds).

type
  Cluster_Atrs = C $\overrightarrow{m}$ A-**set**
  Resource_Inds = Rn $\overrightarrow{m}$ (A $\overrightarrow{m}$ (I × I))
  Atrs_Inds = Cluster_Atrs × Resource_Inds

Requirements Capture 4 *Hence the decision support system for decision support must provide a repository (a data base) for Atrs_Inds as well as appropriate functions, for example for initializing Atrs_Inds, for inserting new, and for displaying, searching, sorting, updating, deleting existing category and resource representor to "actual" attribute sets, respectively attribute and indicator sets.*

5.6 Equity Identification and Definition

Preparation and analysis includes identifying equities and defining a suitable collection of equity functions: their signature (type) and their "behaviour". Some "behaviours" may be only informally defined (**Text**).

type
  Q /* text */
  Equity_Ty = C $\overrightarrow{m}$ (A $\overrightarrow{m}$ I-**set**)
  Equity_Df = E | Q
  Equity_Functs = En $\overrightarrow{m}$ (Equity_Ty × Equity_Df)

Requirements Capture 5 *Hence the decision support system for decision support must provide a repository (a data base) for Equity_Functs as well as appropriate functions, for example for initializing Equity_Functs, for inserting new, and for displaying, searching, sorting, updating, deleting equity function types and definitions.*

In defining equity functions modelling experiments have to be performed in order to establish appropriate models.

type
Type
X_Type = typ_txt:Q × ((Equity_Ty × Type) × Type)
X_Funct = fct_txt:Q × ((RRAIs × VAL) $\xrightarrow{\sim}$ VAL)
Nmd_Xs = Xn $\overrightarrow{m}$ (txt:Q × (X_Type × X_Funct))
X_Res = i_txt:Q × ((RRAIs × VAL) $\overrightarrow{m}$ (r_txt:Q × VAL))
Exec_Xs = Nmd_Xs × (Xn $\overrightarrow{m}$ X_Res)

The experiment functions (hence the use of X) form part of the model being built. They also must first be identified and defined. They finally must be executed and results recorded and annotated.

Requirements Capture 6 *Hence the decision support system for decision support must provide a repository (a data base) for Exec_Xs as well as appropriate functions, for example for initializing Exec_Xs, for inserting new, and for displaying, searching, sorting, updating, deleting experiment function types and definitions. Finally the decision support system for sustainable development must allow execution of the experiment functions.*

5.7 Analysis Function Identification and Execution

Analysis functions are defined in terms of sets, ES, of equity functions. These functions now have to be executed, results recorded and interpreted. Analysis can be viewed as a set of analyses, each named (Lbl), provided with varieties of commented (**Text**) analysis data (RAIs), and with Results also commented (interpreted) and recorded.

Each analysis function argument, besides the rais:RAIS arguments must also be provided with a resource mapping argument. So we need redefine Analysis:

type

iARGS′ = RRAIs × RMs
EARGs′ = iARGS′ × RRAIs
EARGs = {| ((rr,rm),rr′) | ((rr,rm),rr′):EARGs′ • **dom** rr ∧ ... |}
E = (Val × EARGs) $\tilde{\to}$ Fuzzy
ES = En $\overrightarrow{m}$ E
D = RAIs $\tilde{\to}$ RAIs
DS = Dn $\overrightarrow{m}$ D
Analysis = RRS × NmRRAIs × ES
Allocation_and_Scheduling
Plan = Analysis × DS × Allocation_and_Scheduling

Requirements Capture 7 *Hence the decision support system for decision support must provide a repository (a data base) for Plan as well as appropriate functions, for example for initializing, for inserting new, and for displaying, searching, sorting, updating, deleting analyses, including execution of functions and recording results. All insertions and updates usually require the user to provide textual comments (***Text***).*

Executing the modelling and analysis functions require naming the executions:

type
EAn
Exec_Res = (q:Q × (En $\overrightarrow{m}$ (q:Q × (Lbl $\overrightarrow{m}$ (q:Q × VAL)))))
Exec_Plan = EAn $\overrightarrow{m}$ Exec_Res

Requirements Capture 8 *Hence the decision support system for decision support must provide a repository (a data base) for Exec_Plan as well as appropriate functions, for example for initializing Exec_Plan, for inserting new, and for displaying, searching, sorting, updating, deleting analyses, including execution of functions and recording results. All insertions and updates usually require the user to provide textual comments (***Text***).*

We end our example Requirements Capture here as no new principles are being illustrated and as the rest is, from now on, trivial!

5.8 The "Grand" State Σ

Summarizing we can say that the state of a DSS for SD consists of:

type
Σ = Synopsis
× Resource_Info
× Atrs_Inds
× Equity_Functs
× Exec_Xs
× Plan
× Exec_Plan

Requirements Capture 9 *Hence the decision support system for sustainable development must provide a user interface to this state, to its various parts, easy selection and execution of functions: main and auxiliary, user- as well as systems defined.*

5.9 Decision Making

Throughout this and the previous section we have implied that (i) resources had to be identified, (ii) representations sought, (iii) attributes ("of interest") and (iv) indicators (likewise "of interest") had to be determined amongst alternatives, (v) equity and (vi) analysis functions defined, likewise exposing the analyzer and planner to many options. Once analysis functions were executed and (vii) results interpreted choices again arise. Finally when planning, based on analysis, commences (viii) final options present themselves (or otherwise).

All these situations must be carefully recorded; chosen paths (ie. decisions) must also be recorded and it must all be related to the various (i–iix) alternatives.

Requirements Capture 10 *Hence the decision support system for sustainable development must provide easy means for the user: preparer, analyzer and planner, to record all alternatives, to mitivate choices taken, and to "play–back" paths of identification, defintions, executions and choices, also along rejected alternative paths.*

5.10 "The Model"

Throughout this and the previous section we have also implied that a model of the development problem emerges. That model is, in fact, the hypertext–like woven path along alternative and chosen identifications, definitions, executions and interpretations of results and plans.

Requirements Capture 11 *Hence the decision support system for sustainable development must itself, as the user "navigates" around alternatives, selects and rejects choices, etc., build up a graph-like web of the paths taken, with nodes and edges suitably labelled with references to data and functions, explanatory, informal text that the systems elicits from the user, etc.*

We will have more to say about this in section 7.3.

6 A Federated GIS+DIS: $(\mathrm{GaD})^2\mathrm{I}^2\mathrm{S}$

By a **federated geographic information system** we understand a GIS+DIS whose main information, the spatial and (spatially related) census (and other) data and operations over these may reside across a global (i.e. worldwide) network of "ordinary", already established or future geographic information systems "plus" demographic information systems. These latter GISs+DISs may represent their information each in their own way. From a practical point of view such GISs+DISs may be managed on APIC [26], ArcInfo [27], ArcView [28], ERMapper [29], IDRISI [30], InterGraph [31], MapInfo [32], PopMap [33], Redatam [34], and other platforms.

The problem to be dealt with in this section is to properly integrate the concepts of geographic and demographic information systems (GISs, DISs) with the resource representation notions of the previous section.

Hence we need take a look at GISs and DISs. These are aggregations spatially and statistically (tabular) data. By 'federation' we mean the further aggregation of individual GISs and DISs — as they may have been created and are maintained locally, around the world, each covering "separate" but eventually, increasingly more related resources.

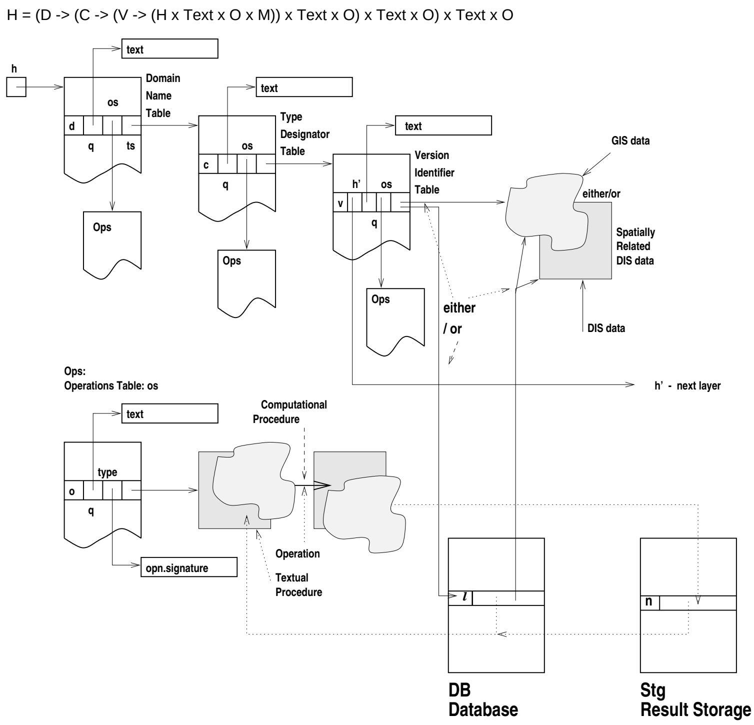
6.1 A First Narrative

To explain our notion of 'federation' further we first present a "picture", then some narrative, and finally a formal model.

A Picture: A *"picture is sometimes worth a thousand words"*. Later we shall claim that a formula is oftentimes worth a thousand pictures.

Yet even pictures need be explained: Squares with emanating arrows designate storage cells of type pointer. "Landscape" rectangles labelled

Fig. 2. A Hierarchical GIS+DIS Federated Information System



'text' designate unstructured, textual data (ie. text which informally describes formatted data — here, from top left to right texts may explain domain, category and version information). "Landscape" rectangles labelled 'opn.signature' designate descriptions of operation types. "Curtain" figures — of which there are two kinds — designate domain/category/version sub-directories, respectively (operation, ie. executable) code directories. The top left to right "curtain" designate sub-directories of domains, categories, and versions (DCV). Directories have fixed format entries ("rows"). Initial DCV sub-directory rows link to further DCV sub-directories. Final DCV sub-directory rows link to either formatted or unformatted data: viz. relations, respectively images — and either directly, the upper arrow, or indirectly, via a database reference, the lower arrow. The final sub-directory rows also links, in the fashion of "recursive descent", to another, lower, layer of domain/category/version directories. The formatted or unformatted data is shown as grey squares or grey "clouds". Code directories link (i) to text briefly explaining the operation, (ii) to the type of the data needed as input to the formally or informally "executed" operation and resulting from those operations, and (iii) to either (formal, executable) code (the arrow-infixed pair of grey "clouds"), or to text (the arrow-infixed pair of grey squares) explaining how the human analyser can go about performing the analysis "by hand"! Performing an analysis function takes input from a database and delivers output to a result storage. Administrative operations, not shown, may move data from the result storage to the database.

6.2 A Second Narrative

Next we reformulate, more systematically, the above symbol explication:

Layers and Levels: Figure 2 displays one layer, with three levels, of a hierarchically structured federated and combined geographic information system and demographic information system: GIS+DIS. (Further layers are referred to implicitly.)

Each of the three large "curtains" (cascaded in the upper left corner of the figure) diagram a table like structure: Domain Name Table (the D level), Category (or rype) Designator Table (the C level), respectively Version Identifier Table (the V level).

Domain Name Tables: One accesses the (or, in general, a) Domain name table from a root, a sentinel, Hierarchy designator (h).

Each entry in the Domain name table contains a distinct domain name (d:D), a (reference to explanatory) text (q:**Text**), (a reference to) an operations table (os), and a (reference to a Category designator table (the latter only shown by an arrow).

One accesses a Category designator table "through" (or via) a Domain name table entry.

Type Designator Tables: Each Category designator table entry contains a distinct type designator (c:C), a (reference to explanatory) text (q:**Text**), (a reference to) an Operations table (os), and a (reference to a) Version identifier table.

One accesses a Version identifier table through a Category designator table entry.

Version Identifier Tables: Each Version identifier table entry contains a distinct version identifier (v:V), a (reference to explanatory) text (q:**Text**), (a reference to) an operations table (os), (a reference to) data, and a (reference to an) [sub-]hierarchy (h).

Data Access: One accesses Data through a version identifier table entry.

Traversing Hierarchy Layers: One also accesses a sub-hierarchy (the "next" layer) through the H item of a version identifier table entry.

Operations Tables: At any D, C or V level one can access an operations table.

Each Operations table (O) entry contains a distinct operations name (on:On), (a reference to) explanatory text (q:**Text**), (a reference to) the type of the operation designated, and (a reference to) the operation [itself!].

Federation means that data may reside on different GIS, DIS, etc, platforms: commercial, experimental, public domain or otherwise: APIC, ArcInfo, MapInfo, IDRISI, PopMap, Redatam, winR+/GIS, etc.

6.3 A Third Narrative

Geographic, Demographic and other Data — Data: We now describe, more generally, but still informally, and from a slightly different viewpoint, the components of the proposed federated GIS+DIS. The base information unit, which usually is a highly composite entity, will be referred to as 'Data'. Examples of specific data are:

Examples 6 *A Geodetic Map of China, A Political Map of Europe, A Vegetation & Natural Crops Map of GuangDong Province (in China), A Mineral Deposits Map of France, A Spatially related Population Census of Zhuhai*[4]*, A Cartographic and Cadestral Map of Macau, etc.*

Domain Names: By D we understand the set of domain names:

Examples 7 *China, GuangDong, Zhuhai, ...*

[4] Zhuhai is a Special Economic Zone of GuangDong Province in China

Data Category Designators: By C we understand the set of composite data types:

Examples 8 *Geodetic Map, Political Map, Vegetations & Natural Crops Map, . . . , Cadestral Map, Population Consensus Data, Import/Export Statistics, . . . , Election Data.*

Version Identifiers: By V we understand the set of version designators (time stamps) of data:

Examples 9 *1982, 1996, . . . , August 3, 1999, . . .*

The Database: Data is kept in a conceptual data base (DB). The data base, as we shall see, can be interpreted as being distributed globally. Each data has a location (L).

Hierarchical Directory Structure: A (d,c,v) identification designates (geographic (GIS), demographic (DIS) and other) data in a hierarchical fashion. Assume fixed type and arbitrary version, then the domain name China could, for example, give access to some data on all of China and then to a set of properly domain-named sub-data of the same (and also other) type(s), etc. One for each (, say) Province of China. And so on, recursively, until some user-defined “smallest grain of data” — which could be a floor plan of a specific residence, a single plot of land for agriculture, etc. This hierarchical (directory-like) recursion is modeled by the below recursion in H.

Data identified “occur” only at the V level of a ‘complete’ (list of one or more) (D,C,V) triples.

Use of the hierarchy (H) entails navigating “up and down” the layers of the hierarchy of (D,C,V) levels. At any one time a user has traversed a Stack of such (d,c,v)’s.

Unary and N-ary Functions — O, On: With each data version there may be some specific, named (unary) functions applicable to that specific data. Designating an operation for application shall mean that the operation is applied to the data designated by the current stack top (which will be a list of (d,c,v) triples — with the list length denoting the current depth of the traversed hierarchy wrt. the root System Hierarchy).

With each specific type we may likewise associate a set of named, unary functions. Each such function is then understood to be applicable to any version of data of that type and domain. The actual data is designated by the topmost stack element whose type matches the operation type.

With each domain we may associate a set of usually n-ary functions. Each such function is then understood to be applied to data designated by the n topmost stack elements whose types matches, in order, the designated operation type.

Some operations may not be computable. Instead text is given which directs the user to "perform" an appropriate evaluation and to enter a resulting value!

Function Result Storage — Stg: Results of operation applications must be uniquely named (by the user) and are stored in a local storage (Stg) under that name together with a historical record of the stack of the time of application, and the appropriately (D,C,V) marked operation name.

At any layer and level domain names, type names, version names, data and operations are annotated by explanatory, descriptive and other text (**Text**).

Operations can be shared across domains, types and versions, as well as across layers of the recursive hierarchy.

Database Sharing and Data Filtering — F: Since also data can be shared across domains, types, versions and layers of the recursive hierarchy, a filter function (F) is provided which, for different levels and layers (etc.) may specialize, generalize or otherwise instantiate the immediately location designated data.

This potentially allows a simple information repository to be viewed, through the (D,C,V) hierarchy as a highly structured (network) of data.

6.4 A Formal (Data Structure) Model

"A small set of formulas is often worth a thousand pictures":

type

$$\begin{array}{l}
\mathrm{D, C, V, Res_Typ, Res_VAL, N} \\
\mathrm{S = H \times DB \times Stg \times Stack} \\
\mathrm{Stack = DCV^*} \\
\\
\mathrm{H = (D \underset{m}{\rightarrow} (C \underset{m}{\rightarrow} (V \underset{m}{\rightarrow} (M{\times}H{\times}Q{\times}O)){\times}Q{\times}O){\times}Q{\times}O){\times}Q{\times}O} \\
\\
\mathrm{M = L \times F} \\
\mathrm{F = Data \stackrel{\sim}{\rightarrow} Data} \\
\mathrm{DB = L \underset{m}{\rightarrow} Data} \\
\mathrm{Stg = N \underset{m}{\rightarrow} (Res_VAL \times Stack \times DCV \times On)}
\end{array}$$

$$\begin{array}{l}\mathrm{DCV} = \mathrm{D} \mid \mathrm{D}{\times}\mathrm{C} \mid \mathrm{D}{\times}\mathrm{C}{\times}\mathrm{V} \\ \mathrm{OTup} = ((\mathrm{A}^* \times \mathrm{C}^*) \times \mathrm{Res\_Typ}) \\ \mathrm{OFct} = (((\mathrm{VAL}^* \times \mathrm{Data}^*)|\mathrm{Q}) \stackrel{\sim}{\rightarrow} \mathrm{Res\_VAL}) \\ \mathrm{O} = \mathrm{On} \underset{m}{\rightarrow} \mathrm{OTyp} \times \mathrm{OFct} \times \mathrm{Q}\end{array}$$

Yet even the formulas may have to be narrated — and that was done in the three Sections 6.1—6.3.

6.5 Data Sharing, Viewing and Gluing

The indirect reference, via M, in the database DB to the geographic information system or demographic information system Data is provided for a number of reasons:

Local Layering: For each layer descending M's (i.e. L's) may refer, in fact, to "overlapping" (probably embedded) Data. At one (say an "upper") layer an L refers to a "large" spatial area (or a large census table), whereas at a "lower" the L may refer to an "smaller" area probably properly contained in the "larger" area. The View functions F therefor serve to sub-locate the right sub-Data!

More concretely: If a domain name at an "upper" layer is 'Europe' then through the recursive decent through some (C,V) designated H we get the domain names: 'Denmark', etc. The "upper" L designated perhaps a map of Europe, whereas the "lower" should designate a map of Denmark.

Quite specifically: In a Cartographic & Cadestral Service the maps of a city may be in the database DB as a set of "gluable" sub-maps. These may cover areas not related to administrative or other domain nameable entities. The various layers now "zoom" in on successively "smaller", but administratively "well-rounded" areas. The purpose of the view functions are to collect from one or more sub-maps Data covering the located area and "glue" it together.

Global Distribution: The database may itself be distributed — and across the globe! Now L's (with their F's, i.e. the M's) also contain for example INTERNET information (etc.) so that the Data can be located "in somebody else's database"!

6.6 A Relational View

In the presentation of the Federated GIS+DIS given so far we may have left the reader with the impression that access to the global information is through a strict sequence of triples of domain, then type and finally version identifiers.

We now lift this seeming restriction to allow for a relational access approach. Instead of the (d,c,v)-list view so far proposed and formalized:

type
$H = D \underset{m}{\overrightarrow{\rightarrow}} (C \underset{m}{\overrightarrow{\rightarrow}} (V \underset{m}{\overrightarrow{\rightarrow}} (M \times H \times ...) \times ...) \times ...) \times ...$

we instead suggest a relational view:

type
rH
RelH = (rH × H)**-set**
H = D × C × V× rH × O × Q

rH is like a relation tuple identifier.

It is easy to see that any relation RelH can be mapped into either of:

type
$H = D \underset{m}{\overrightarrow{\rightarrow}} (C \underset{m}{\overrightarrow{\rightarrow}} (V \underset{m}{\overrightarrow{\rightarrow}} (M \times H \times ...) \times ...) \times ...) \times ...$
$H' = C \underset{m}{\overrightarrow{\rightarrow}} (D \underset{m}{\overrightarrow{\rightarrow}} (V \underset{m}{\overrightarrow{\rightarrow}} (M \times H \times ...) \times ...) \times ...) \times ...$
$H'' = V \underset{m}{\overrightarrow{\rightarrow}} (C \underset{m}{\overrightarrow{\rightarrow}} (D \underset{m}{\overrightarrow{\rightarrow}} (M \times H \times ...) \times ...) \times ...) \times ...$
etc.

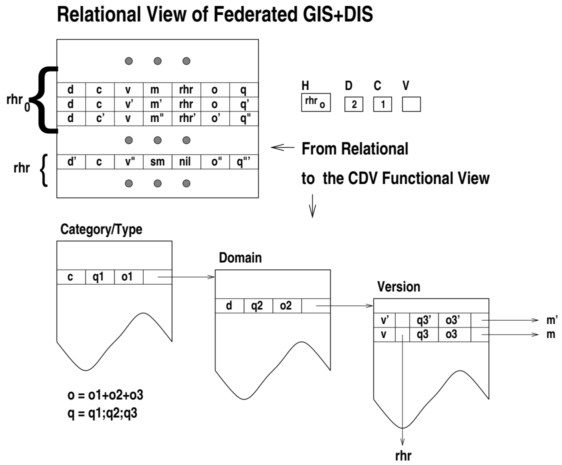
Given a relational representation the user can then determine, at any layer to view the information base, which ordering of the (d,c,v)'s to select — and the system can respond by presenting the tables as selected.

Initially the system "sets" the hierarchy layer (H), for example: rhr_0. Subsequently the user sets, in sequence two of either of the D, C, or V "buttons".

7 A GIS+DIS–based DSS for SD

In the decision support system for sustainable development we dealt with resources, with representations of resources, with attributes and indicators, and with functions over resources and resource representations, attributes and indicators.

Fig. 3.



7.1 Spatial Resource Maps and Filters

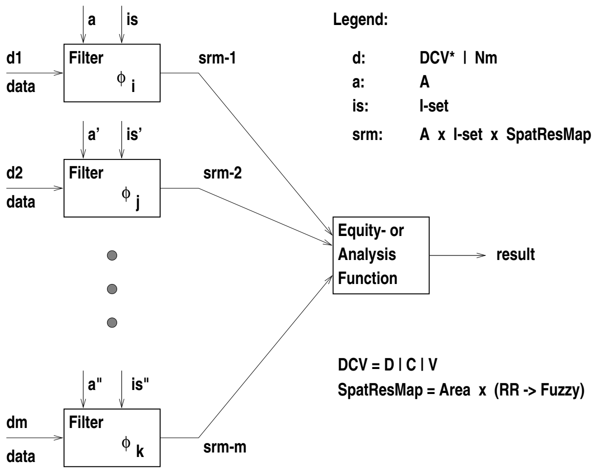
With respect to spatially related resources, we do not record the individual resources or their representations. Instead we typically, when it comes to for example environmental resources, record highly complex aggregations of numerous such resources in the form of for example remotely sensed images.

From these we are then, somehow, able to extract, or as we shall call it: filter, representations of resources, one-by-one. Typically, however, the (for example) remotely sensed data also contains a confusing aggregation of other data that somehow must be screened away.

type
Φ
Coordinate = **Real** $\times$ **Real** $\times$ **Real**
Area = Coordinate-**set**
SpaResMap = Area $\xrightarrow[m]{}$ (RR $\xrightarrow[m]{}$ Fuzzy)
AIs = A $\times$ **I-set**
Filter = (AIs $\times$ Data) $\stackrel{\sim}{\rightarrow}$ (AIs $\times$ SpaResMap)
Filters = Φ $\xrightarrow[m]{}$ Filter

So what we have, usually in a geographic information system are maps, or images, of complex aggregations of Data, and what we want are sim-

Fig. 4. A Generic Spatial Resources Map and its Filters



ple recordings, in the form of well-defined Spatial Resource Maps of resources. By a Spatial Resource Map, we understand a mapping from a an area, that is: a set of three dimensional coordinates to a map from Resource Representations to Fuzzy qualifiers. The idea is that the spatial map "cleanly" represents only those resources for which certain attribute values are present and within given indicator ranges. We choose to map from an area in order to capture averaging properties. Thus a Filter is a function from a triple of Attribute designators, Indicator ranges and (for example an image of remotely sensed) Data to a Spatial Resource Map.

Several filter functions usually are needed to prepare input for the Equity and Analysis functions:

Requirements Capture 12 *A GIS+DIS–based DSS for DS must therefore allow the preparer, analyzer and planner to develop, record and apply filter functions (Φ).*

7.2 The "Grand" System

The Data provided to the Filter functions "come" from the $(\mathrm{GaD})^2\mathrm{I}^2\mathrm{S}$ repositories: either accessed through an appropriate DTV name list or by the name of a stored result.

This basically completes the GIS+DIS–based DSS for SD System description.

Requirements Capture 13 *A GIS+DIS–based DSS for DS must therefore allow the preparer, analyzer and planner to "link" up the* DSS for SD *resource concepts with the Data concepts accessible through the recursive hierarchy of domain, type and version names and through the names of results stored, with comments, after human evaluation or computed execution of Equity, Analysis and Planning functions.*

7.3 Towards "The Model"

Very briefly: the hyper-text "woven path" also includes the generation of graphs like the below:

Fig. 5. Towards a Development Model

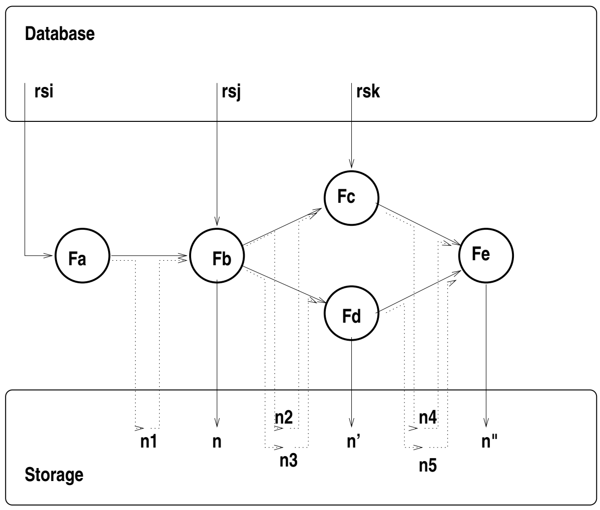


Figure 5 shows a number of Analysis functions and their interrelations with respect to input and output data. Output from some function applications serve as input to other function applications. Outputs (results) are named, and so are input arguments. The above graph (albeit conceptual and very simple) shows an overall "functionality", an overall "structure" of the problem, one that is often, in the demographic information system literature, referred as being "ill-defined" and "unstructured"! In the above picture we have simplified many aspects: simple provision of resource arguments (rather than their prior filtering through filters etc., no user provided invocation time arguments, etc.

Requirements Capture 14 *A GIS+DIS–based DSS for DS must therefore be able to draw, upon request from the preparers, analyzers, planners, developers, and decision makers, the "model" graph of all Functions invoked — whether their results were ever again applied or not — together with a complete "trace" of Data used, whether originating from the Database or from Storage (where it would reside if that Data was the result of previous Function applications).*

8 Conclusion

8.1 On DSS for SD

We have sketched a main outline of how we intend to tackle the issue of decision support system for sustainable development, for how we intend to tackle the issue of a federated geographic information system and demographic information system, and for how we intend to combine them into $(GaD)^2I^2S$: a Federated GIS+DIS DSS for SD.

We have separated two concerns: the DSS for SD from the Federated GIS+DIS. And then we have combined them.

We usually find the issue of DSS for SD "cluttered up" by the mixing of problems of for example deciphering what spatial maps contain of information and the "pure" issues of resources, their attributes, indicators and equity functions. So we have separated the two issues. To then make the whole separation work we bring the issues together.

8.2 On Software Development

We have sketched main phase and some techniques of a new approach to software development: One that is based on domain models on which requirements are then based, and on requirements models on which software development is then based.

This approach is currently under intense research, development and application [3–9].

But much work need to be done before we can fully justify our claims: we need now carefully study relevant papers. The study will emphasize our "isolation" of the resource, attribute, indicator, equity function etc. issues in order to validate sections 4–5. The paper studies will then analyze the issues of geographic information system and demographic information system functionalities in order to validate section 6.

Meanwhile we will be "building" a "prototype" $(GaD)^2I^2S$ to make more precise the requirements Capture items mentioned in sections 5–7, and to check the conceptual elegance, consistency and comprehensiveness of the $(GaD)^2I^2S$ proposal.

8.3 Acknowledgements

I thank (i) participants in the February 1996 UNU/IIST Workshop on Software Technology for Agenda'21: Decision Support Systems of Sustainable Development [35, 36] for instigating the modelling effort of this paper and for kind remarks, (ii) my colleagues at UNU/IIST for enabling me to enjoy five stimulating years as first and founding UN director of that flourishing UN University software technology research and post-doctoral training centre in Macau, (iii) members of the IFIP WG 2.2 and IFIP WG 2.3 Working Groups for providing stimulating critique of my work, and (iv) Hans Langmaack for 20 years of close friendship.

References

1. The RAISE Language Group. *The RAISE Specification Language*. The BCS Practitioner Series. Prentice-Hall, Hemel Hampstead, England, 1995.
2. The RAISE Method Group. *The RAISE Method*. The BCS Practitioner Series. Prentice-Hall, Hemel Hampstead, England, 1992.
3. Dines Bjørner. Domains as Prerequisites for Requirements and Software *&c.* In M. Broy and B. Rumpe, editors, *RTSE'97: Requirements Targeted Software and Systems Engineering*, volume 1526 of *Lecture Notes in Computer Science*, pages 1–41. Springer-Verlag, Berlin Heidelberg, 1998.
4. Dines Bjørner and Jorge M. Cuellar. The Rôle of Formal Techniques in Software Engineering Education. *Annals of Software Engineering*, 1999. Editors: Norman E. Gibbs and N. Coulter.
5. Dines Bjørner. Where do Software Architectures come from ? Systematic Development from Domains and Requirements. A Re–assessment of Software Engneering ? *South African Journal of Computer Science*, 1999. Editor: Chris Brink.
6. Dines Bjørner. Pinnacles of Software Engineering: 25 Years of Formal Methods. *Annals of Software Engineering*, 1999. Editors: Dilip Patel and Wang YingYu.
7. Dines Bjørner. Domain Modelling: Resource Management Strategics, Tactics & Operations, Decision Support and Algorithmic Software. In J.C.P. Woodcock, editor, *Festschrift to Tony Hoare*. Oxford University and Microsoft, September 13–14 1999.
8. Dines Bjørner et al. Formal Models of Railway Systems: Domains. Technical report, Dept. of IT, Technical University of Denmark, Bldg. 344, DK–2800 Lyngby, Denmark, September 23 1999. Presented at the FME Rail Workshop on Formal Methods in Railway Systems, FM'99 World Congress on Formal Methods, Toulouse, France. Avaliable on CD ROM.

9. Dines Bjørner et al. Formal Models of Railway Systems: Requirements. Technical report, Dept. of IT, Technical University of Denmark, Bld.g 344, DK–2800 Lyngby, Denmark, September 23 1999. Presented at the FME Rail Workshop on Formal Methods in Railway Systems, FM'99 World Congress on Formal Methods, Toulouse, France. Avaliable on CD ROM.
10. John Fitzgerald and Peter Gorm Larsen. *Developing Software using VDM–SL*. Cambridge University Press, The Edinburgh Building, Cambridge CB2 1RU, England, 1997.
11. Gro Harlem Brundtland, editor. *Our Common Future*. World Commision on Environment and Development. Oxford University Press, WCED, UN, 1987.
12. UN. Agenda'21. United Nations, The Rio de Janeiro, Brasil, Conference on Environment, June 14 1992.
13. LI Xia and Anthony Gar-On YEH. A dss for sustainable land development in china using remote sensing and gis — a case study in dongguan. In *[35]*, 1996. Centre for Urban Planning and Environmental Management + GIS/LIST Research Centre, University of Hong Kong, Pokfulam Road, Hong Kong; hdxugoy@hkucc.hku.hk.39.
14. International Union for the Conservation of Nature. World conservation strategy. Technical report, International Union for the Conservation of Nature, Gland, Switzerland, 1980. Report highlights sustainability of natural resources.
15. A.G.Levinsohn and S.J. Brown. Gis and sustainable development in natural resource management. In M. Heit and A. Shrtreid, editors, *GIS Applications in Natural Resources*, pages 17–21. GIS World, Inc., 1991.
16. Anthony Gar-On YEH. Gis in decision support systems for sustainable development. In *[35]*, 1996. Centre for Urban Planning and Environmental Management + GIS/LIST Research Centre, University of Hong Kong, Pokfulam Road, Hong Kong; hdxugoy@hkucc.hku.hk.23.
17. U.E. Loening. Introductory comments: The challenge for the future. In A.J. Gilbert and L.C. Braat, editors, *Modelling for Population and Sustainable Development*, pages 11–17, London, England, 1991. Routeledge.
18. A. Steer and W. Wade-Grey. Sustainable development: Theory and practice for a sustainable future. *Sustainable Development*, 1(3):223–35, 1993.
19. C. Ponting. Historical perspectives on sustainable development. *Environment*, (4-9):31–33, November 1990.
20. L.K. Caldwell. Political aspects of ecologically sustainable development. *Environmental Conservation*, 11(4):299–308, 1984.
21. M.R. Redclift. *Sustainable Development: Exploring the Contradictions*. Methuen, London and New York, 1987.
22. S.R. Dovers and J.H. Handmer. Contradictions in sustainability. *Environmental Conservation*, 20(3):217–222, 1993.
23. Nicholas Georgescu-Roegen. *The Entropy Law and the Economic Process*. Harvard University Press, Cambridge, 1971.
24. Julian Simon. *The Ultimate Resource*. Princeton University Press, Princeton, N.J., 1981.
25. B.J. Brown, M.E. Hanson, D.M. Liverman, and R.W. Meredith Jr. Global sustainability: Toward definition. *Environmental Management*, 11(6):713–719, 1987.
26. Apic. Apic news: Le journal d'apic systems. Technical Report 6, Apic Systems, 25, rue de Stalingrad, F-94742 Arcueil, Cedex, France, 1995.
27. Environmental Systems Research Institute (ESRI). *Understanding GIS: The ARC/INFO Methods*. Number Version 7 for UNIX and Open VMS. GeoInformation International, 307 Cambridge Science Park, Milton Road, Cambridge, CB4 4ZD, United Kingdom, 3 edition, 1995.

28. Environmental Systems Research Institute. Introducing arcview. Manual, Environmental Systems Research Institute (ESRI), ESRI Inc. 380 New York Street, Redlands, California 92373-2853, USA, 1994.
29. Earth Resource Mapping. Er mapper 5.0 – product information. Literature with demo version, Earth Resource Mapping, Level 2, 87 Colin Street, West Perth, Western Australia, 6005, 17 January 1995.
30. J. Ronald Eastman. Idrisi for windows. User's Guide Version 1.0, Clark Labs for Cartographic Technology and Geographic Analysis, Clark University 950 Main St., Worcester, MA 01610-1477 USA, May 1995.
31. INTERGRAPH. Geographic information systems. Product info. folder, Intergraph Corporation, One Madison Industrial Park, Huntsville Albama 35807-4210, USA, 1995.
32. MapInfo. Mapinfo reference. Reference manual version 3.0, MapInfo Corporation, One Global View, Troy, New York 12180-8399, 1994.
33. UNSTAT/DESIPA. Popmap: Integrated software package for geographical information, maps and graphics databases – user's guide and reference manual. Technical Report ST/ESA/STAT/107, Department for Economic and Social Information and Policy Analysis, Statistical Division (UNSTAT), United Nations, New York, New York 10017, USA, 1994.
34. CELADE/ECLAC. Redatam-plus version 1.1. User's Manual Distr. GENERAL: LC/DEM/G.90 Series A, Nr. 201, Latin American Demographic Centre (CELADE)/United Nations Economic Commission for Latin American and Caribbean (ECLAC), Casilla 91, Santiago, Chile, December 1991.
35. Dines Bjørner, Zbigniew Mikolajuk, Mohd Rais, and Anthony Gar On Yeh, editors. *Decision Support Systems for Environmentally Sustainable Development — Software Technology for Agenda'21*, UNU/IIST, P.O.Box 3058, Macau, February 25 — March 8 1996. IDRC (International Development Research Centre, Ottawa, Canada) and UNU/IIST (United Nations University, International Institute for Software Technology), UNU/IIST. Unpublished Workshop Hand-outs. (Workshop co-sponsored by IDRC: The Canadian Governments' Intl. Devt. Research Centre, Ottawa.)
36. P.A.V.Hall, D.Bjørner, and Z.Mikolajuk (eds.). Decision Support Systems for Sustainable Develpoment: Experience and Potential. Position Paper 80, UNU/IIST, P.O.Box 3058, Macau, August 1996. International workshop on *Decision Support Systems for Environmentally Sustainable Development — Software Technology for Agenda'21* co-sponsored by IDRC: The Canadian Governments' Intl. Devt. Research Centre, Ottawa.

Laudatio:

Hans and Annemarie Langmaack have been friends of us for now 20 years. My wife and I treasure this friendship.

Hans has been my colleague for a bit more than 20 years. And I am very grateful for this.

I first met Hans though IFIP Working Group 2.2: Formal Description of Programming Concepts, in the ancient imperial and breat city of Kyoto, a hot and humid August 1978. Hans invited me later to become the visiting professor at the Danish chair at

Christian Albrechts University of Kiel, the second oldest university of Denmark when Schleswig-Holstein was Danish, till 1864. My lectures at Kiel can be said to have had som influence. But I sincerely believe that that influence was due more to Hans' insistance and wisdom than to my lecturing. In any case: One thing led to another. Hans' group at Kiel became engaged, also, in research into, development of tools for and applications of formal techniques in software development: Notably VDM.

Later Hans and his group was a natural choice when, at the instigation of Tony Hoare, the **ProCoS** *project consortium was founded — and with Hans followed, also a happy choice, Ernst-Rüdiger Olderog, then on his way from Kiel to Oldenburg.*

Here at Kiel we all met the best of the central European academic tradition: Fine professors, deeply engaged in the well-being of science and of their students, a cultural life, albeit, as most such life was and is in Germany — a heritage from its many dukedoms and princely states before the world went awry — provincial, with music: Playing at home, as does Anne-Marie, local concerts, and the province festival; theatre and opera, when in season, outings to nearby seats of local gentry, historical studies, and to the 'Freilichts Museum': The great collection of farm buildings and farming equipment, reminicent of times gone, and also reminicent of the fine heritage that was Hans': His upbringing in West Frisia.

Now an academic era comes to an end, in one person. But true to Hans' devotion: He has brought up almost two generations of scientists who will take over, here i Northern Germany, as well as elsewhere in Germany. They will bring forward his always gentle engagement, his deep concern for what really matters in computer and computing science and in software engineering.

Hans from one Dane to almost another Dane — and to Anne--Marie, from me and Kari to You and Anne-Marie: Thanks for a wonderful summer of 1980, and for so many happy moments together: Here, im frischen wind, around other parts of your Germany: Oberwolfach and Dagstuhl, in Kyoto, Macau, Lyngby, Oxford, and many other places. Thanks for your steadfast leadership and love of your field of study, your students and your friends from across the sea.

Table of Contents