# Domain Modelling: Strategic, Tactical and Operational Resource Management

Dines Bjørner

**Abstract**

In this contribution to the symposium in honour of the Work of Tony Hoare, marking his retirement from Oxford University, we examine and illustrate programming methodological issues of domain modelling — an activity we see as logically, if not temporally, preceeding those of requirements engineering and software design.
We identify such method principles and techniques as domain perspective, intrinsics, support technology, rules & regulations and human behaviour modelling.

The main body of the paper is an example. That example allows us to illustrate four different things: (i) domain modelling, (ii) strategic, tactical and operational resource management, (iii) an interface between computing science and operations research, and (iv) a spectrum of software: From decision support software to algorithmic software.

## Laudatio:

Tony Hoare has influenced my scientific work and technological outlook perhaps more than most.

His ability to formulate method principles and techniques, his quest for laws of programming, at all levels, not just for imperative and parallel programming, but also for abstract specification and for development, and his ability to lead you gently into these and other subjects through the classical mastery of the English language, never ceases to amaze me: Truly a man for many more seasons !

I first met Tony at the 1966 NATO Summer School in Villard de Lans, near Grenoble, as a listener; then through triennial meetings at the Oberwolfach Mathematics Centrum in the German Black Forest (from the late 1970s to the mid 1980s), and through my observer and subsequent membership of the IFIP working group WG 2.3 on Programming Methodology (from 1979 onwards). It was at a meeting in the Belgian Vallon that Tony, always kind and overbearing, after an Edsgerian[1], blistering — but no doubt well-deserved — critique of a *presentation of mine of a topic for discussion,* that Tony sitting down, asking to write in my note book, first said (something like): *If we are to tackle the issue of correctness of systems, then we must secure that not only the problem specification, its requirements and the software relate correctly, but also*

[1]Edsger Wyse Diijkstra

*that the compiler, the run-time, incl. operating systems, and the hardware machine is similarly correct — and: Most notably, we must be able to formulate this system correctness issue on less than one sheet of normalsize paper. Otherwise we stand no way of comprehending the issue.* As a result we created the **ProCoS** project — and met many times. I was, do I need to say this ?, quite happy to be able to introduce my colleagues — some my former students, and some of my then students: Kirsten Mark Hansen, Michael Reichhardt Hansen, Hans Henrik Løvengreen, Anders Peter Ravn, Hans Rischel, Erling V. Sørensen, etc. — to Tony. As a result of this EU ESPRIT BRA sponsored project, I was also able to attract Zhou ChaoChen to join us, for three years, both at Lyngby and at Oxford. It is a sign of the wisdom and foresight of Tony that the **ProCoS** project became a fine scientific success !

In [1] Gries writes of Hoare: "Tony likes to write. To write and rewrite."

Thanks Tony for showing us the fruits of your labour and for inviting us all in to try do likewise, and through that, for enjoying our labour so even more ! Thanks.

`Inside every large software system`
`there is a little program desparately trying to get out !`

C.A.R. Hoare

## 1 Introduction

Large, man-made systems such as transportation (railways, air traffic, etc.), health care, public administration, and manufacturing (construction, production, etc.), can benefit from computing support. But sometimes their software contains too many "desparate programs".

The idea of domain modelling: Of doing one's utmost to more thoroughly understand such application domains as mentioned above, before any attempt is made at formulating requirements for software — let alone designing that software — that idea, may be a way to ensure that resulting software systems embody "harmonious, content and happy" programs.

That is: We find that domain careful domain modelling may enlighten the reqirements engineers in such ways that resulting software is "fit for purpose".

The brillant ideas that past software engineers might have had, and which lead to their being "programs", often were "botched" by subsequent "improvement, generalisation, 'add-on-feature', etc." software extensions (by other, or even the same, software engineers), which then led to the "original programs" becoming "desparate": The initial concepts "suffocated" from lack of elegance etc.

In this paper we shall show some of the 'sketching' that seems to enhance the fostering of robust ideas and hence survivable programs.

### 1.1 "All we do is 'Describing' "!

Simplifying somewhat: *All we do, in software development, is describing.*

From the development, as we shall see in this paper, of domain understanding, via the specification of requirements, to the design of software, whether it be of an architecture (which emphasises the software properties as externally observable), or of the program organisation (which emphasises the internal interfaces), or of subsequent designs — from the domain, via the requirements to the software — we describe.

Although it is by now a trivial observation, it cannot be repeated too often in a world of science, still dominated, it seems, by natural science researchers and engineering technologists: And the last step of these descriptions, that we expect to be carefully, painstakingly — and often formally — related, the last step of these descriptions is the product. No drilling, honing, planing, turning, etc.; no digging, pouring, laying, assembly, etc. The transition from the description of the world as it is[2], in the domain, via the world as we would like it to be (as in requirements), to the world as it will become (as in the software), those transitions we must strive to make as smooth and obvious as possible.

And since we describe, and since descriptions are meant to be read, all of them, many stages and steps, by people, it is likewise important, to us, that these descriptions, fruit of mostly intellectual labour, are beautiful: Are elegant and captures our attention[3] !

In the engineering of mechanical artefacts, of building structures, of electronics, we primarily apply measures of quantity: Our artefacts usually each cover, in comparison, to the software systems of the earlier quote, narrow physical phenomena. In the engineering of software, and in particular the kind of "all-encompassing" systems, some development issues, of which this paper will address some, in such development one is faced with the reconciliation of many, broader issues: Satisfying oftentimes dozens of rather distinct groups of stake-holder perspectives, mediating between many technological, system management rules & regulation, and human behaviour facets. The software is not bound by laws of the natural sciences, as are the material quantity-measured systems referred to above, but basically by laws of mathematics and, secondarily, by the "laws" of the man-made "systems" of the infrastructures in which the software is to serve. As a result, it seems, our measures of product success are measures of intellectual quantity: Correct, pleasing and beautiful.

It is on this background: Of descriptions and of intellectual qualities of both the design process and of the resulting product, that the reader is invited to see the particular focus, paid in this paper, on domain engineering.

### 1.2 A programme of research and education

For centuries science has studied physics and chemistry, zoology and botanics, geology, etc. The natural phenomena.

For many years computer science has studied semantics of programming and of specification languages: Man-made, computing-centric phenomena.

[2]Here it is a pleasure to also recognise the influence that Michael A. Jackson has had on our work [2].

[3]Here it is a pleasure to also recognise the influence that Edsger W. Dijkstra has had on our work [3].

This paper suggests that it is high time to also study the semantics of large--scale man-made systems. If we ever are to conquer the complexity of software systems, if we ever are to make end-users confident in our systems, and if ever these man-made systems are to be made manageable: "Cut down to human size", then we must, it is strongly believed, study these man-made systems, notably the very large-scale infrastructure components listed earlier. Not just as is done, for narrow, isolatable fragments of for example the scheduling and allocation of resources, as is done in operations research. Not just as is likewise done for similarly system identifiable, monitorable and controllable parts of engineering systems, as is done in control theory. But for as encompassing, wide and deep perspectives and facets of human systems: Air, land and water transport (air traffic, airports, air lines, railways, buses, taxis, road (in general), shipping, . . . ), financial services (banking, insurance, securities, portfolio management, instrument clearing, . . . ), fisheries industry, health care (rural paramedics, clinics, hospitals, medical insurance, healthy and sick people, . . . ), manufacturing (market analysis, design, production, sales, service, . . . ), commerce (trading: buying/selling — producers, suppliers, consumers, traders — the market), etc.

*Why is it that this must be done ?*

Well: Software systems are today being used and built, systems which — in a usually unreflected manner — are gathering diverse program packages into such "wholes" which, more-or-less, are "encrouching" upon these large-scale infrastructure components. They arise, without much reflection: That is, little if any attempt has been made to 'create' models, and hence theories about such systems. (Recall the citation of a "Tony'ism"!)

*Why must we, in computing science, do this ?*

We have the means, at present, more than any other scientific discipline, for creating these models. As the development of models, ie. informal and formal, reflected, descriptions of these infrastructure components — as these developments — become common and accepted, further research and development will undoubtedly be taken over by "natural habitats" of these systems: Transportation engineering, departments and institutes of finance, the health care industry, etc.

Computer and computing science have, in a unique way — "foreign" to other sciences and the humanities — researched and developed (linguistic) means of description: Abstract specification and concrete programming, the logics and algebras behind these, and, in general, computable models of well-nigh anything: Logic, algebraic and concurrent calculi. Where control theory deployed traditional differential calculi in modelling continuous and chaotic behaviours, computer science can combine these with discrete event and abrupt state behaviours.

So for a time computing science shall probably point the way !

### 1.3 Structure of paper

To bring home this manifesto: That computing science must, among its many activities, include such which systematically create models of large-scale, man-made infrastructure components, we shall bring a single example — that of resource management at the strategic, tactical and operational level.

## 2 The example

This example arises out of an analysis of two kinds of enterprise sectors: manufacturing and transport, that is, production and service enterprises. Instead of analysing the domain of a particular industry, or business, and abstracting from it (the domain analysis, the industry cum business), we show here an example which has singled out a common trait of all such enterprises, and then purports to analyse this trait across "all" such industries and businesses. The trait is that of *resources* and their *management*.

This example shows the ability of formal specification to capture the essential differences between an enterprise's three, albeit pragmatic levels of resource management. The strategic management of resources: Their "down-sizing" (sell-off, divestment) or "upgrading" (acquisition, investment); the tactical management of (physically relocatable [mobile]) resources: Their spatial allocation and scheduling; and the operations management of resources: Their allocation to tasks and the scheduling of these tasks. At the same time we are able to "put our finger" on what distinguishes 'algorithmic' software from 'decision support software'.

In the first sections which follow we shall analyse the notions of resources, locations and time [4].

## 3 Resources: "A data model"

In a transportation or a manufacturing business we single out resources, their allocation and scheduling — the latter to satisfy some time-table requirements — as main objects of concern.

We take a conventional space-time approach to allocation and scheduling. Thus we equate allocation with the assignment of a set of spatial locations to a resource, and scheduling with the assignment of a set of time intervals to a resource. The former may be the locations where the resource is either being applied or idled. The latter time intervals may be those during which the resource is captured for use by a specific task, or is available for capturing. Thus we shall also deal with notions of time intervals and tasks.

### 3.1 Resource component types

We focus initially on the resource components. At the end of this section we will then briefly review functions applicable to resources.

We characterize components in terms of their types and the values that a component can attain wrt. a given type.

• Resource States:

We say that values contribute to the 'state' of the component.

Examples of state parts (of components) are given below and relate to the:

- 'primary, **pragmatic** purpose': from item 1 on the following page to item 6 on page 7;

  or to more **semantic** properties such as:

- 'capacity': item 7 on the facing page;
- 'usability', from item 8 on page 8 to item 10 on page 8;
- 'mobility', from item 11 on page 8 to item 13 on page 8;
- 'activity', from item 14 on page 8 to item 16 on page 8;
- 'allocatability & schedulability', from item 17 on page 9 to item 18 on page 9;
- 'quality', from item 19 on page 9 to item 23 on page 9;
- 'maintenance', from item 24 on page 9 to item 26 on page 9;
- 'locality', from item 27 on page 9 to item 29 on page 9;
- 'ownership', from item 30 on page 10 to item 32 on page 10; and
- 'authorisation', from item 33 on page 10 to item 34 on page 10.

So, at any one time a resource is in a state. That state has many parts. These are of the categories for example listed above, and to be exemplified below. Each part has types and values. Some values are (static) constant, others are (temporal) variable. A resource can thus be described by the type of its overall composition (of parts) and all its (recursively descending) sub-types.

We observe that we do not treat time and locations as resources, or that we at least single them out for special treatment.

- Example Resource Type Classes:

Examples, that is: types of resources include those of their 'primary, pragmatic purpose':

1. *Cash, Monies, Capital, . . . :*
   - **Airline:** *Stock holders' and venture capital's investments, passengers' and cargo freighters' payments, bank credits, etc.*
   - **Manufacturing:** *— same as for the* Airline Company *example.*
2. *Human staff:*
   - **Airline:** *Management (headquarter, logistics, operations, etc.), administrative staff (personnel, financing, accounts, procurement, etc.), aircraft crew (pilots, cabin personnel), ground staff (check in, gate, tarmac, etc.), sales & reservations staff, etc.*
   - **Manufacturing:** *Management, and administrative, inventory & warehouse, marketing, services and sales, design, shop floor, maintenance, etc. staff.*

3. *Facilities:*
   - **Airline:** *Sales and check-in counters, passenger lounges, gates, aircraft parking lots, luggage conveyor belts, runways, etc.*
   - **Manufacturing:** *Head office, design office, production line, warehouse and other buildings.*
4. *Main production equipment:*
   - **Airline:** *Aircrafts.*
   - **Manufacturing:** *design office product development tools (CAE, CAD, etc.); shop floor machining tools (lathes, honers, planers, cutters, etc.) and moving equipment (conveyor belts, fork lifts, trucks, etc.); inventory and warehouse mechanisms (stacks, fork lifts, etc.); etc.*
5. *Materials:*
   - **Airline:** *Fuel, air ticket and airway bill forms, meals, drinks, etc.*
   - **Manufacturing:** *Raw materials (ingots, metal blocks, sheets, rods, etc.), parts (atomic [bolts, nuts, washers, etc.], semi-assembled [chassis, motor block, transmission, doors, seats, dash board, etc.]; etc.), etc.*
6. *Auxiliary equipment:*
   - **Airline:** *Office equipment (desks, filing cabinets, computers, communications, etc.), spare parts, etc.*
   - **Manufacturing:** *As for Airlines.*

### 3.2 Resource component sub-types

We abstract, in this example, from many aspects of resources. Categories of resources were mentioned, in detial, above. Except, perhaps, for monies, we did not go into properties (types and values for such types) of resources. A further development (analysis and modelling) of resource management would have to go into such attributes:

7. *Capacity:*
   Any resource has a 'capacity' for performing tasks.
   The notion of capacity is perhaps the most important one in connection with production resources.
   - **Airline:** *Aircraft passenger and/or cargo load and volume, flying distance, maximum fuel load, etc.; staff: number of working hours per day, week and month.*
   - **Manufacturing:** *Shop floor machinery: number of items processable per hour, day and week; etc.*

Capacity is usually a constant property of a given resource.
A third way of looking at resources groups them wrt. 'usability':

8. *Consumable resources: cash, materials, etc.*
A consumable resource is one which when spent, ie. applied, by a task, is "lost": Cannot, as 'that' resource, be recuperated. See analysis (*&c.*) Section 5.1.
   - **Airline:** *Monies, fuel, meals, time.*
   - **Manufacturing:** *Raw and other materials, oil for operating the production machinery.*
9. *Serially reusable resources:*
A serially reusable resource is one which after having been captured and used, ie. after having been released, by one task, by a task can (later) be reused by another (or the same) task. See analysis (*&c.*) Section 5.1.
   - **Airline:** *Staff and aircraft.*
   - **Manufacturing:** *Staff, design and production machines.*
10. *Shared resources:*
    - **Airline:** *Ticket and check-in counters, office equipment, administrative staff, etc.*
    - **Manufacturing:** *Sales outlets, etc., as for airlines.*

We may further view resources, wrt. 'mobility', as being either:

11. *Fixed, immovable:*
    - **Airline:** *Runways, gates, ticket and check-in counters.*
    - **Manufacturing:** *Design office equipment, manufacturing shop floor machinery.*
12. *Locally movable:*
    - **Airline:** *Ground staff, passenger and cargo vehicles, tow-away trucks.*
    - **Manufacturing:** *Production line workers, materials, trucks.*
13. *Globally (nationally, regionally, continentally, etc.) movable:*
    - **Airline:** *Aircrafts, aircraft crews.*
    - **Manufacturing:** *Marketing, services and sales staff, products.*

Resources are bound, 'activity'-wise:

14. either waiting to be used, ready to be captured
15. or being used productively,
16. or waiting for or under repair.

We may not need to distinguish between the latter two ‘states’: items 15 and 16.

17. a resource can always be allocated and scheduled.
18. and at any one time it is either allocated, or scheduled, both or none!

in either of the above three ‘activity’ states.

That is we allow the possibility of reserving resources for future productive uses or for maintenance, where the allocations and schedulings may take place while the resources in question are actually being ‘applied’.

A resource is always in a certain ‘quality’ state:

19. in “tip-top’ shape, ready for or being used (ie. applied)
20. in operable shape but “nearing” a need for being diagnosed w.r.t. a possible need for preventive maintenance (up-keep)
21. ready for preventive or adaptive maintenance
22. unsafe for use
23. broken down — and thus in need of corrective maintenance

Thus:

24. *Preventive maintenance:* is an operation that usually requires equipment resources and is performed on resources to secure the future error-free operation of the latter resources.
25. *Adaptive maintenance:* is an operation that usually requires equipment resources and is performed on resources to up- or down-grade their functionality or operability, including capacity.
26. *Corrective maintenance:* is an operation that usually requires equipment resources and is performed on resources to repair faulty (broken) resources.

Resources may be accessed:

27. *In situ* — on the spot for immovable resources:
    - **Airline:** *Runways, gates, counters.*
    - **Manufacturing:** *Design office equipment, shop floor machinery.*
28. *Locally:*
    - **Airline:** *Ground staff, passenger and cargo vehicles.*
    - **Manufacturing:** *Production line workers, trucks.*
29. *Globally:*
    - **Airline:** *Aircrafts, sales office staff.*
    - **Manufacturing:** *Marketing, service and sales staff.*

The latter two forms (items 28 on the page before and 29 on the preceding page) may give rise to a notion of distribution.

Another way of looking at resources is provided by the following 'ownership' distinctions. Resources are either:

30. *Internal:*
    ie. owned fully by the enterprise for which we are implementing a model.
    - **Airline:** *Cash, own staff, non-leased aircraft, etc.*
    - **Manufacturing:** *Cash, own staff, non-leased equipment, etc.*
31. *External:*
    ie. accessed (and even transformed) by that enterprise which, however, has no %'age of ownership in the resource.
    - **Airline:** *usually airport facilities such as lounges, gates, runways.*
    - **Manufacturing:** *Industrial waste sewage treatment, public utility supply, etc.*
32. *Partnership:*
    ie. partly owned by the resource.

depending on whether the enterprise being domain analyzed controls, or do not control the resource.

Ownership, as above, is an issue of legal, ie. property ownership. But it begs analyzing a derived facet, namely access rights.

Resource access rights may be granulated:

33. no restriction
34. access restricted only to certain resources (or other?)
35. *&c.*

Access rights is an issue strongly related to tasks. See analysis (*&c.*) Section 5.1.

### 3.3 Resource component functions

In a summarising sense we may be able to claim that there are the "objects" that are the resources, there are their locations in space, and there are the times when they occupy or are otherwise "destined" for those locations or for tasks involving the resources.

Disregarding the temporal and spatial properties, that can be associated with resources, we can subject them to (other) functions that concern the non-tempral, non-spatial attributes.

For example:

36. Resources can be created: *Clients pay for products (goods, services or transportation). Cash is being replenished through sale of stocks or bank credits or loans.*
37. Resources can be transformed: *cash resources are converted into for example equipment (through paying purchase price) or staff (through salary).*
38. Resources can be allocated and scheduled.
39. Resources can be captured or released (idled, made available).
40. Captured resources can be applied to other resources — or even clients — with the latter being transformed (while the former are being applied):
41. The state of a resource can be inquired.

- **Airline:** *An aircraft and its crew is being applied to a number of passengers (clients) and these passengers as well as the resources are being transformed: moved in space!*
- **Manufacturing:** *Machining tools are being applied to materials and yield products (resources).*

Application, by tasks, of resources to other resources can be for purposes of for example either productive use or for reasons of preventive, corrective or adaptive maintenance. See paragraph of that subject page 17.

### 3.4 Resources: A summary

- **Definition:** By a resource we understand a uniquely identified component which has a great variety of attributes (the type and subtypes of the resource), with values, and which can be subjected to various observer and generator functions.

The type, sub-type and function subsections were rather sketchy. A thorough treatment is warranted. To do so — for a given application domain — we would make use of classification techniques from "ontological computer science". [5, 6, 7, 8] are but a few referenes with which we are familiar. This domain knowledge formalisation sub-field is under intense study.

In summary we can say that with every resource there is a typically partially ordered set (a lattice) of types, and a number of resource observer and generator functions:

**type**
  R_type
  OBS = R → VAL
  GEN = R → R
**value**
  anal: R → R_type–**poset**

We shall not be explicitly using these functions, but they are needed in order to detail the resource planning functions of section 6.

# 4 Spatial and temporal operations on resources

We will focus on the spatial and temporal relations that resources can enter into [4].

We will not define actual operations, such as:

**value**
allocate: R $\times$ ... L ... $\rightarrow$ ...
schedule: R $\times$ ... T ... $\rightarrow$ ...
allocsch: R $\times$ ... T $\times$ L ... $\rightarrow$ ...

but we will hint at the structures (...) that arise from these operations.

## 4.1 Locations and allocation

Analysis

The concept of [spatial] allocation, in this section, is that of assigning a location to a resource, or a set of such locations. Our view here is perhaps not the traditional one. In a computing system we speak of *processes being allocated to processors.* Here we speak of *resources being allocated to locations!* We shall later examine these two views in some detail. See (task etc.) analysis Section 5.1.

Examples of spatial resource [al]locations are:

42. Fixed locations
    - **Airline:** *Airports, their runways, the ticket and check-in counters, the gates and conveyor belts of an airport have distinct, fixed locations.*
    - **Manufacturing:** *Design offices, machining and assembly lines of a manufacturing enterprise have distinct, fixed locations.*
43. Sets of local locations
    - **Airline:** *Locally movable airline equipment may be stored (when idled, ie. not scheduled (not in use)) in any of a number of not necessarily related locations.*
    - **Manufacturing:** *Same as for an airline.*
44. Global locations
    - **Airline:** *Air routes*
    - **Manufacturing:** *Sales network*

Conclusion

- **Location:** Locations are presently further unanalysed entities.
- **Resource allocation:** By a single resource space allocation we mean the association of a spatial location with a single resource.

In summary:

**type**
  L
  RA = R $\overrightarrow{m}$ L

### 4.2 Time and scheduling

Analysis

It is usually a good idea to avoid modeling time till as late as possible in computable models. Since, however, we are not [yet] concerned with computable models [of software], and since the "reality" that we are dealing with rather explicitly mentions time we have little choice but to model time. Concretely, time can be described in terms of various forms of dates and clock times. With t:T we designate a time t, in the domain of T, such that T's elements all embody syntactically consistent notions of calendar year, month, week, day in week and in month, hour, minute, second, etc. — where 'etc.' refers to arbitrary fractions of seconds should that be necessary.

Time intervals play an important rôle: the times from and to, ie. the times between begin and end times. A task must occur within a given time interval, or: a task must begin no earlier than a certain time and no later than a certain other (later) time where the latter may be much later, or very soon thereafter: ie. "hard real-time"!

Functions and operations performed on resources take time to execute. We model by an interval the minimum to maximum expected or actually observed times it take to execute functions and operations.

Resources are present in some enterprise over some interval of time, and, within such intervals they can be allocated and scheduled, respectively idled and captured for properly embedded and non-overlapping (ie. disjoint) time intervals. The concept of schedule is that of assigning time intervals with a resource. Speaking of a time interval in which a resource is scheduled shall mean that the resource is 'bound'. The resource may be bound to being available for anyone to capture for deployment or that it is bound to not being so available. So we may then introduce subsidiary notions of *capturing* and *releasing* such resources over certain time intervals. A resource which is not scheduled basically does not exists! Thus acquiring new resources shall mean that they are generally scheduled to be either available or occupied within the overall time interval of the resources being acquired by an enterprise (bought or rented) — that is: till the last time they can be deployed. Thus we may speak of a strategic schedule which expresses which resources an enterprise may operate and over which (usually time-wise large) intervals. Within a strategic schedule the bound resources may now appear in tactical and/or operational schedules as either being available or reserved.

We now bring the two independent concepts: allocation and scheduling together. The idea of combining spatial allocations over time intervals with resources is simple: namely that of expressing that a resource is required to be physically present in a certain location during a certain time interval.

- **Airline:** *For a flight to be dispatched a designated aircraft must be physically present in the departing airport for some time before the scheduled departure time.*
- **Manufacturing:** *For the manufacturing of a certain product to start constituent parts must be physically available at the production site immediately prior to scheduled production.*

The above example hinted at some static presence. The next examples hint at some dynamic "presence".

- **Airline:** *For a flight to occur a designated aircraft must be physically present in the air-lane during the interval of time of flight.*
- **Manufacturing:** *For the manufacturing of a certain product to take place it, and its supply parts must be physically available along the production (machining, assembly) line during the scheduled production.*

Conclusion

- **Definitions:**
  - **Time:** By time we understand an abstraction that embodies the year, month, week, day, hour, minute, second, etc., of the 'time' spoken of!
  - **Schedule:** By a single resource schedule we mean the association of a time interval with a resource.
  - **Allocation and Schedule:** By a resource allocation and schedule we understand an association of a time interval and a space of locations with a resource.

Scheduled availability of resources (SAR) (in certain time intervals), and scheduled spatial allocation (SSA) can be modelled as:

**type** R, L, T
  SAR = T×T $\overrightarrow{m}$ R-**set**
  SSA = T×T $\overrightarrow{m}$ (R $\overrightarrow{m}$ L)

#### Abstract time

Time, as an abstract concept — and time, when "implemented inside" the computer, become an abstract concept — gives rise to both philosophical and logical considerations [9]. We shall not venture into these here, but only point out that a proper model of any domain in which time plays a crucial rôle need consider it axiomatic foundation carefully. Essentially that calls for some form of temporal logic being introduced [10, 11, 12, 13, 14, 15, 16, 17, 18].

## 5 Resource contexts

Resources, their scheduling and allocation, is motivated by production and service tasks in response to customer orders. In preparing for resource planning, past production statistics and estimates of future orders must be available.

### 5.1 Production, servicing, tasks

Usually one thinks of resource allocation and scheduling as the binding (association) of locations (spaces) and time intervals to resources with respect to a given task. A task is a set of actions usually performed in a given time interval. The purpose of the task is to deliver a set of products. The view of bindings of resources to locations and time intervals given so far is a "macro-view": that is, a view at a high level as opposed to a "micro-view".

Strategic, tactical and overall resource planning seems to assume the "macro-view" of resource allocation and scheduling enunciated so far. They consider the entire operation of the enterprise as one "grand exercise" in resource allocation and scheduling. Operational planning, in contrast, may take the "micro-view" of resource allocation and scheduling, now binding tasks to resources!

- **Airline:** *Flying passengers and transporting cargo are the main products of an airline. Thus an offer to fly a passenger between two locations is a commitment to deliver a product. Having commenced the journey the product can be considered delivered!*
- **Manufacturing:** *Manufactured goods such as a car, a bread toaster or a semi-assembled and machine product such as a door assembly, or a spare part for a bread toaster — all are examples of products. Servicing a car or a bread toaster are also examples of products.*

- **Airline:** *The process of flying passengers and/or cargo is a task; so is informing a potential passenger of time-table and selling them a ticket.*
- **Manufacturing:** *The process of designing a product is a task. So is the process of machining or assembling parts, or both!*

The bindings of the above exemplified tasks to resources, where these are allocated to locations and otherwise time constrained (ie. resource scheduled), are now exemplified.

- **Airline:** *The 'flying passengers and cargo' task is bound to an aircraft resource which again has been scheduled. The 'informing about and selling of tickets' is bound to a set of sales agents, that is a set of resources which is otherwise bound to sets of locations and scheduled.*
- **Manufacturing:** *The process of designing, or producing a product is bound to a number of design office, respectively shop floor resources, each of which are allocated to locations and scheduled.*

Spatial resource allocation and scheduling permits the scheduling of tasks. When the number of spatially resources is small compared to the number of tasks desired, then the binding of tasks to resources may become a critical issue.

One view of what is 'real' about real time is that a task may have to meet a completion deadline in a context of limited resources.

- **Definitions:**
  - **Products:**
    By a product we mean a resource.
  - **Product Description:**
    By a product description we understand an annotated, acyclic graph. Labeled nodes of the graph designate typed actions. Labeled directed edges of the graph designate flow of resources between nodes.

Node and edge labels can be bound to resource types and quantities.

Some (input) arcs are directed at nodes but do not emanate from other nodes. They designate points where resources "flow into" the denoted task.

Some (output) arcs emanate from nodes but are not directed at other nodes. They designate points where products "result" from the denoted task.

Some (input) arcs are directed at nodes but do not emanate from other nodes, but from "local storage".

Some (output) arcs emanate from nodes but are not directed at other nodes, but from "local storage".

- **Definitions — Continued:**
  - **Production Allocation & Scheduling:**
    By a production allocation and scheduling we understand a product description all of whose nodes and arcs are bound to scheduled and spatially allocated resources.

A task is, loosely speaking, a set of one or more sequences of functions and operations applied to (task-bound) resources to produce a product. A task thus represents an "execution" of an allocated and scheduled product description. Execution of a task must occur in the time intervals for which the respective resources are scheduled.

- **Definitions — Continued:**

- **Tasks & Task Traces**
  By a task we understand a production allocated and scheduled product description and a partial, continuous function — called the task trace — which to any time in an interval assigns a set of node and arc labels.

The idea is the following: if at time t functions and operations are bing applied (to material) at nodes ns and are flowed via arcs ws, then the task trace maps t into the union of ns and ws.

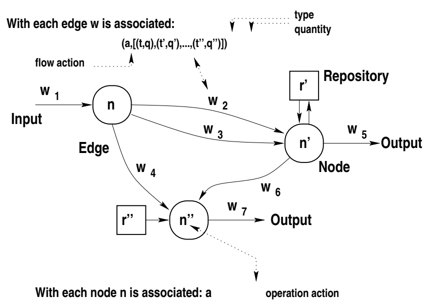
### Production [or Servicing] Plans and Tasks

: Production of goods or services thus follow certain acyclic, graph-like production plans where nodes (N) designate actions (A, like machining or service rendering) and where arcs (W) designate quantified (**Nat**) flow (also a kind of action, A) of typed (RTyp) resources. Each kind (P) of product (or service) plan has its own production plan (PP). External input (inputs) initiate production, external output (outputs) deliver results. At each node there may be many next (NXT) nodes and a local repository of material resources (REP).

Figure 1 shows a production plan.

**type** P, Pn
**axiom** P $\subset$ R
**type** N, W
  PPs = (Pn $\overrightarrow{m}$ PP)
  PP = inputs:NXT
    $\times$ graph:(N $\overrightarrow{m}$ (NXT $\times$ REP))
    $\times$ outputs:NXT
    $\times$ actions:((N|W) $\overrightarrow{m}$ A)
  NXT = N $\overrightarrow{m}$ (W $\overrightarrow{m}$ (RTyp $\overrightarrow{m}$ **Nat**))
  REP = RTyp $\overrightarrow{m}$ **Nat**

Figure 1: Production Plan

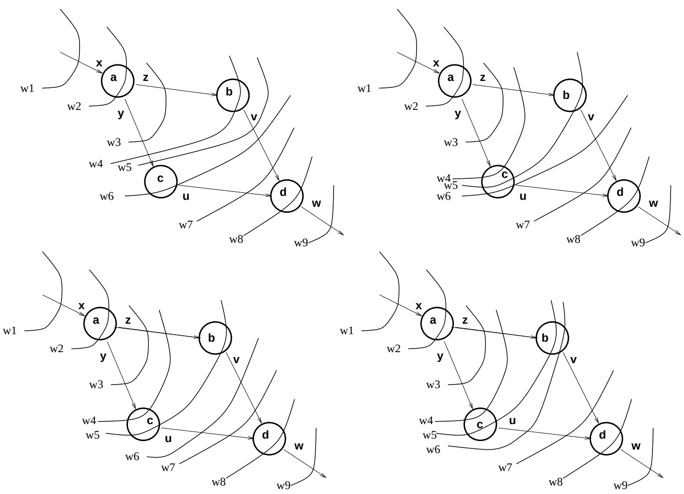


Product Tasks & Traces

: Production (or servicing) resources are needed in order to carry out actions (A). These are (to be) task scheduled and allocated to nodes and edges (NWAS) for [each] product plan (ASPD[s]). The dynamics of a task is now modelled as a pair: the (static) task scheduled and allocated plan and a (the dynamic) trace. For each instance of time the trace records at which nodes and on which edges the locus of control (of executon of the task) resides. A task thus records the past history and planned progress of the production, or servicing of one order.

Figure 2 shows four possible traversals of a production graph.

Figure 2: Production Traces




**type**

NWAS = (N|W) $\overrightarrow{m}$ ((T×T) $\overrightarrow{m}$ (R $\overrightarrow{m}$ L))

ASPD = PP × NWAS

ASPDs = Pn $\overrightarrow{m}$ ASPD

Task = ASPD × Trace

Trace = T $\stackrel{\sim}{\to}$ (N|W)-**set**

Tasks = T $\stackrel{\sim}{\to}$ (Pn $\overrightarrow{m}$ Task-**set**)

Figure 2 shows four traces:

⟨{x},{a},{y,z},{b,y},{y,v},{d,v},{u,v},{d},{w}⟩

⟨{x},{a},{y,z},{c,z},{b,c},{c,v},{u,v},{d},{w}⟩

⟨{x},{a},{y,z},{c,z},{b,c},{u,v},{d},{w}⟩

⟨{x},{a},{y,z},{c,z},{b,c},{b,u},{u,v},{d},{w}⟩

The node and edge actions avail themselves of the node, respectively edge scheduled and allocated task resources.

At any moment an enterprise is honouring, for each named product, zero, one or more productions (servicings), ie. tasks.

### 5.2 Clients and orders

The transportation as well as the manufacturing industry exists because there are clients, ie. people and institutions who wishes to be, respectively have their staff or cargo, transported, or who wishes to procure products.

The transportation as well as the manufacturing industry exists because there are clients, ie. people and institutions who wishes to be, respectively have their staff or cargo, transported, or who wishes to procure products.

Clients, as a concept, is considered orthogonal to the concept of resources.

- **Airline:** *Passengers, freighters and cargo.*
  *Passengers inquires about travels and reserve, modify, consume and cancel tickets. A passenger who consumes a ticket is moved from one location to another. Cargo requested transported by a freighter. Initially the freighter interfaces with the airline, and then for a long interval. Within the interval the cargo "takes over". Finally the freighter ("at the other end") receives (has the) cargo (unloaded).*
- **Manufacturing:** *Buyers.*
  *A buyer inquires about products, specifies and orders specially described or ordinary, off-the-shelf, receives, accepts and pays for products.*

- **Definition: Client:**
  By a client we understand an object who is not a resource but who otherwise is subject to resource consuming and enriching operations.

Clients place orders to be delivered during various time intervals. Orders lead to production.

- **Airline:** *A future passenger buys tickets for a number of specific flights at specific times.*
- **Manufacturing:** *A client orders various manufactured goods to be delivered over time.*

- **Definitions: Orders**
  An order (for example) specifies the client, the ordering time, the product(s) being ordered and time intervals during which specified quantities of (each of) the product(s) shall be delivered. An order is identifiable through a unique order number.

Production (etc.) occur in response to, or expectation of clients (C, placed, placeable) orders. An order (O) names a product (resp. service, Pn), the time interval (ti) in which the order is to be delivered and a quantity (q). Each client order receives a unique order name (On).

**type** C, On
  O = Pn $\overrightarrow{m}$ (ti:(T×T) $\overrightarrow{m}$ q:**Nat**)
  COs = C $\overrightarrow{m}$ (On $\overrightarrow{m}$ O)

### 5.3 Objectives, statistics and estimates

Strategic planning involves establishing objectives. Objectives reflect past performance (ie. statistics), current strength and expected (ie. estimated) production, that is: business.

Hence we must analyze the concepts of objectives, performance and estimates.

An objective calls for certain resources to be present or to have been generated (met) during, or over, certain time intervals.

- **Airline:** *Over a certain time interval and within a certain geographical area (whose local airport is served by the airline in question), that airline expects to sell and is thus willing to fulfill a number of seats for each of a variety of flights.*
- **Manufacturing:** *Over a certain time interval and within a certain geographical area (which is served (including serviced) by the manufacturing enterprise in question), that enterprise expects to sell and is thus willing to fulfill quantities of orders for each of a variety of products.*
- **Definitions:**
  - **Objective:**
    By a business objective we mean a listing which for a number of resources, including products, list quantities that are expected in 'stock', respectively sold, at, respectively in a certain area.
  - **[Performance] Statistics:**
    By a statistics we mean a table which for various time intervals and locations lists the actually achieved quantities for each resource.[4]
  - **Estimate:**
    By an estimate we mean we mean a table which for various time intervals and locations lists the business objectives.

A business objective (BO) records for suitable recources the number required. The number may be negative! Statistics (S) and estimates (E) records for suitable time intervals and locations the business objectives for that time interval and at that location.

[4] ie. a table identical in form to an estimate.

**type**
  BO = R $\overrightarrow{m}$ **Int**
  S,E = (T×T) $\overrightarrow{m}$ (L $\overrightarrow{m}$ BO)

Statistics records past experience. Estimates record future expectations.

# 6 Resource management

## 6.1 Strategic planning: sp

We recall:

**type**
  sar:SAR = (T×T) $\overrightarrow{m}$ R-**set**
  ssa:SSA = (T×T) $\overrightarrow{m}$ (R $\overrightarrow{m}$ L)

which expresses either a desired or an actual state: in some (planned or actual) interval some (planned or actual) resources are available (or occupied, or to be scheduled) (SAR), respectively at some (planned or actual) locations (SSA).

At some time, t, not shown, we wish to obtain a set of possible future resource suggestions covering an interval $(t', t'')$ and for given past performance (ie. statistics s) and future estimates, e, which in total must cover the embedding time interval (t′,t″).

The "function" sp generates a set, sars, of suggested resource schedules for that interval.

Each member of the set (sars) lists for appropriate time intervals, ($t_i$,$t_j$) properly within (t′,t″), the 'build-up' or the 'down-sizing' of resources based on the available resources, rs, the statistics, s, and the estimates, e.

We suggest that sp delivers a set of alternative target plans. Some target plans may involve a demand on "externally owned" resources that cannot be honoured by the market. Which such, external resources can be honoured by the market can only be determined after negotiation. It is therefore important that a number of alternative such target plans can be contingency prepared for any negotiation process.

- **Airline:** *An airport may not be able to offer desired landing rights. An air traffic control authority may not be able to offer desired fly-through options.*
- **Manufacturing:** *Parts suppliers may not be able to deliver desired quality and quantity. Electric power companies may not be able to provide required peak power. Investment capital may not be able to provide sufficient capital within desired time frames.*

Other target plans may offer alternative self-owned, ie. internal resource build-ups which may be subject to internal negotiation with for example stock-holders. As the model for the latter closely follows that of a model for the former we omit further treatment of this case.

Ownership of resources can be graduated (ie. granulated) by the resource type mechanism.

Projects: Meta-Production:

We wrote above: 'the "function" sp'. sp is not expected to be a computable function, let alone an algorithmically executable procedure. But sp's main components can — most likely, by any reasonably mature enterprise — be reasonably precisely described, albeit informally.

Just like we can present procedures for the production of products, so, of course, we can describe procedures for the "production" of plans like the targets. Thus we can expect to use the same mechanism for the production of products as for the planning of strategies, tactics, etc.

This observation gives rise to a distinction between the concepts of production, respectively projects:

**Production** designate the operational handling of clients and products.

**Projects** designate the planning (market analysis for, and research and development) of new products, as well as the planning of the strategies, the tactics and the operations!

The clients see the results of production, but is hopefully spared insight into underlying projects — whereas the enterprise staff is most often more concerned with the projects than with the production!

- **Planning:**
  Strategic planning is based on available resources, statistics and estimates, and yields, for (future) time intervals plans for possibly downsized or upgraded resources. There may be an infinity of "solutions" to strategic planning.

  **value**
  sp: R-**set** $\times$ S $\times$ E $\stackrel{\sim}{\to}$ (T$\times$T) $\stackrel{\sim}{\to}$ SAR-**infset**
  sp(rs,s,e)(t$'$,t$''$) **as** sars
  **pre** ... **post** ...

  We have assumed that this form of strategic planning toccurs in "isolation" from the market.

- *Negotiation:* ng **Analysis:** — Ends on page 23
  Let b:B designate the identity of the enterprise for which strategic planning is being performed.
  The negotiation process now "pitches" a number of enterprises against one another and, collectively, against the external market. The negotiation process results in the market offering a guaranteed build-up or maintenance (or even down-sizing) of "externally owned" resources over various time intervals. The offer is accepted by the enterprise, b, in question (ie. the enterprise conducting the above negotiation with the market) — that

is: in the context of competing enterprises, each with their unique identity (ie. designator b:B).
The negotiation process is usually an iterative one in which negotiation phases, that is applications of 'ng', alternates with simulated applications of the next two functions: resource reconciliation and estimate revisions.

**End of** *Negotiation:* ng **Analysis**

- **Negotiation:**
  Out of such a possible infinity of choices (or zero, for that matter!) one has to be chosen, if possible. Now negotiationtakes place in the context of the market. Each "player" (B) has own, strategic resource plans.

  **type** B
  **value**
  ng: (B × SAR-**infset**) × (B $\underset{m}{\rightarrow}$ SAR-**infset**) $\overset{\sim}{\rightarrow}$ SAR
  ng((b,sars),bmsars) **as** sar
  **pre** ... **post** ...

- *Reconciliation:* rec **Analysis:** — Ends on page 23
  The offered and (tentatively) accepted, or potentially acceptable "external" target plan is used as basis for selecting an appropriately "matching internal" target plan from the set of original, alternative, possible target plans. That is: a "merge" of some sar$''$ from sars with sar yields, approximately, a (resolved) sar$'$ (perhaps in sars).

  **End of** *Reconciliation:* rec **Analysis**

- **Reconciliation:**
  Negotiation results, seen from each "player" in a possibly negotiated (ng) resource schedule which may have to be reconciled with those (sp) originally planned, resulting a yet a possibly other, new resource schedule.

  **value**
  rec: E × sp:SAR-**infset** × ng:SAR $\overset{\sim}{\rightarrow}$ result:SAR
  rec(e,sars,sar) **as** sar$'$
  **pre** ... **post** ...

- *Revision:* rev **Analysis:** — Ends on page 23
  Given that agreements have (now) been made with the market concerning the temporal access to external resources, the enterprise (in question) usually need adjust its original estimates. Only such estimates which can be supported by available (temporal access to) external resources (in addition to own, internal resources) are relevant.
  The "function" rev generates one such compatible and revised estimate.

  **End of** *Revision:* rev **Analysis**

Finally strategic planning may thus revise original estimates.

- **Revision:**

  **value**
  rev: E × SAR $\stackrel{\sim}{\to}$ E
  rev(e,sar) **as** e′
  **pre** ... **post** ...

  – **Interpretation:** The "functions" sp, neg, rec and res are just given their signature. A number of **pre/post** conditions can be expressed — typically after some thorough knowledge-engineering acquisition — but not enough for us to be able to actually compute definite, let alone all answers. Together with the signatures and the **pre/post** conditions one can, however, formulate *requirements* to a (so-called "expert-system"-like) **decision software support system**.

### 6.2 Tactical planning: tp

Based on expected (ie. estimated) and actual orders and on available resources we can now establish the spatial allocation and scheduling of resources necessary for a subsequent orderly production.

- **Airline:** *A product may be a flight. To each flight there corresponds then a set of two or more airport locations each with their arrival and departure times (ground stop intervals).*
- **Manufacturing:** *A product may be a set of one or more automobiles. To each such set there is a 'schedule' for their initial and final availability in a number of locations.*

Tactical resource management thus allocates scheduled resources in the context of estimates and actual (or expected, current) orders:

**value** tp: E × COs → SAR $\stackrel{\sim}{\to}$ SSA

- **Interpretation:** Remarks similarly to those above for the strategic "functions" can be attached to this tactical "function", and again we should be in a position to *knowledge-engineer*, ie. *requirements capture* a *decision support sosftware system.*

### 6.3 Operations planning: op

Whereas tactical planning primarily developed plans which, when carried out, secures spatial presence (ie. location) of resources during required time intervals, operational planning shall develop plans which, when carried out secures that operations can indeed proceed smoothly: that is, that tasks are allocated to resources! Operational management now takes current (and immediately expected future) orders and spatially allocated and scheduled resources and allocates them to production (or servicing) tasks:

**value** op: COs × SSA $\stackrel{\sim}{\to}$ ASPDs

- **Interpretation:** The "closer", however, we get to operations, the more our planning functions become algorithmically well-defined. Although one may not arrive at optimal algorithms for the implementation of *operations management* (due to NP completeness), this function is typically realisable using modern operations research analysis techniques.

Operations : The production plans can now be executed:

**value** ops: APSDs $\stackrel{\sim}{\to}$ Tasks

- **Interpretation:** The argument to this operations despatch, monitoring and control function, *ops*, and desired **pre/post** conditions, are such that this function is (almost trivially) algorithmically computer supportable.

Discussion This example shows: (i) that it is possible to make reasonably non--trivial statements about strategic and tactical resource management and (ii) to relate these to operational management, (iii) that we can model at least three stake-holder perspectives within a "smooth" spectrum of models for: (1) strategic (owner and top-level executive) management, (2) tactical, divisional level management, and (3) operations management, and (iv) that we can identify a spectrum of software from *knowledge based decision support software* to *operations research analysis based* task scheduling and production monitoring & control *software*.

## 7 Conclusion

The principles and techniques of domain engineering, as sketched in this paper, has been further discussed and studied in [19, 20, 21, 22, 23] and is a main subject of the authors' lecture notes [24].

## References

[1] David Gries and Fred B. Schneider. *A Logical Approach to Discrete Math.* Springer–Verlag, 1993.

[2] Michael Jackson. *Software Requirements & Specifications: a lexicon of practice, principles and prejudices.* ACM Press. Addison-Wesley Publishing Company, Wokingham, nr. Reading, England; E-mail: ipc@awpub.add-wes.co.uk, 1995. ISBN 0-201-87712-0; xiv + 228 pages.

[3] W.H.J. Feijen and E.W. Dijkstra, editors. *Beauty is our business : A birthday salute to Edsger W. Dijkstra*, Texts and monographs in computer science, New York, N.Y., 1990. Springer–Verlag.

[4] Wayne D. Blizard. A Formal Theory of Objects, Space and Time. *The Journal of Symbolic Logic*, 55(1):74–89, March 1990.

[5] Jørgen Fischer Nilsson. An Algebraic Logic for Concept Structures. In H. Jaakkola et al., editor, *Information Modelling and Knowledge Bases*, Amsterdam, The Netherlands, 1993. IOS Press.

[6] Jørgen Fischer Nilsson and Jari Palomäki. A Categorical View on Concept Structures. In H. Kangassolo et al., editor, *Information Modelling and Knowledge Bases*, pages Chapter 16, 239–256, Amsterdam, The Netherlands, 1995. IOS Press.

[7] M. Ingleby. A galois theory of local reasoning in control systems with compositionality. In *Proceedings of Mathematics of Dependable Systems.* Oxford UP (UK), 1995.

[8] Jørgen Fischer Nilsson. Object Logic for Conceptual Modelling — with a Medical Domain as Case Study. In Y. Tanaka et al., editor, *Information Modelling and Knowledge Bases*, pages 330–343, Amsterdam, The Netherlands, 1996. IOS Press.

[9] Johan van Benthem. *The Logic of Time*, volume 156 of *Synthese Library: Studies in Epistemology, Logic, Methhodology, and Philosophy of Science (Editor: Jaakko Hintika).* Kluwer Academic Publishers, P.O.Box 17, NL 3300 AA Dordrecht, The Netherlands, second edition, 1983, 1991.

[10] J.M.E. MacTaggart. The unreality of time. *Mind*, pages 457–474, 1908.

[11] A.N. Prior. *Past, Present and Future.* Clarendon Press, Oxford, England, 1967.

[12] P. Röper. Intervals and Tenses. *Journal of Philosophical Logic*, 9:451–469, 1980.

[13] B. Moszkowski. *Executing Tempral Logic Programs.* Cambridge University Press, Cambridge, England, 1986.

[14] E. Harel, O. Lichtenstein, and A. Pnueli. Explicit Clock Temporal Logic. In *Symposium on Logic in Computer Science*, pages 402–413. IEEE, IEEE CS Press, 1990.

[15] Zohar Manna and Amir Pnueli. *The Temporal Logic of Reactive Systems: Specifications.* Addison Wesley, 1991.

[16] Zhou Chaochen, C.A.R. Hoare, and A.P. Ravn. A Calculus of Durations. *Information Processing Letters*, 40(5):269–276, 1991.

[17] Zhou Chaochen. Duration Calculi: An Overview. In *Proceedings of Formal Methods in Programming and Their Applications, D. Bjørner, M Broy, and I.V. Pottosin (Eds.)*, pages 256–266. LNCS 735, Springer-Verlag, 1993.

[18] Zohar Manna and Amir Pnueli. *The Temporal Logic of Reactive Systems: Safety*. Addison Wesley, 1995.

[19] 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.

[20] 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.

[21] 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.

[22] Dines Bjørner. Pinnacles of Software Engineering: 25 Years of Formal Methods. *Annals of Software Engineering*, 1999. Editors: Dilip Patel and Wang YingYu.

[23] Dines Bjørner. A Triptych Software Development Paradigm: Domain, Requirements and Software. Towards a Model Development of A Decision Support System for Sustainable Development. In *Festschrift to Hans Langmaack, Eds. Ernst-Rüdiger Olderog and Bernhard Steffen*, volume 1710 of *Lecture Notes in Computer Science*, Heidelberg, Germany, October 1 1999. University of Kiel, Germany, Springer–Verlag.

[24] Dines Bjørner. Software Engineering: A New Approach — from domains via requirements to software architectures and program structures. Lecture notes, Department of Information Technology, Technical University of Denmark, Bldg.343, DK–2800 Lyngby, Denmark. Web URL: http://www.it.dtu.dk/˜db/s2000/notes.ps (presently some 700 pages) [foils: .../foils.ps], 1999.