# Laws of Man-made Systems

## Software Support for Infrastructures

## Invited Lecture: CARI'96

## Libreville, Gabon, 10 October 1996

Notes for Interpreters, Projectionist and Speaker + Foils

# Opening

## Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

Foil 1 on projector 1.

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

First I would like to express the sincerest thanks on behalf of the UN University and in particular on behalf of UNU/IIST: the UNU's International Institite for Software Technology, to the organizers of this event, its sponsors, and amongst them notably the IAI: the African Institute for Informatics, here in Lireville, and INRIA, France for inviting the UNU and UNU/IIST to become part of CARI'96. Thank you.

In this talk I will introduce you to a concept of Infrastructures. I will present informal as well as formal models of examples of Infrastructures and dwell, briefly on Laws of such Infrastructures. There will be many examples, some 10 if we get time to show them all, and there will be seemingly complicated mathematics. But do not worry: the mathematics is really not the essence of this talk. Instead the essential features of this talk are: (i) Semantics of man-made Systems, that is: Infrastructures; (ii) An attempt to formally define the concept of Infrastructures; and to show Laws of Infrastructures.

# Laws of Man-made Systems

## Software Support for Infrastructures

1. The Infrastructure Concept
2. Small Models of Large-Scale Infrastructures
3. Laws of Man-made Systems
4. Research & Development Issues:
   (a) Theories of Infrastructures
   (b) Domain Analisys vs. Requirements Capture
   (c) Software Architecture vs. Program Organisation
5. UNU/IIST in a Developing World

**Dines Bjørner**

**Chris George, Tomasz Janowski, Richard Moore, Søren Prehn**

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

Foil 2 on projector 1.

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

So: a major emphasis of this talk is that of giving semantics to man-made infrastructure systems — such as listed here.

We see an infra-structure system syntactically as consisting of a set of sub-languages and semantically as their universes of discourse. So we give semantics to selected terms— not to arbitrary sentences — of such languages .

At the end of the talk we will put the technical parts of the talk in the context of research and development at UNU/IIST and possibly at Academgorodok (Moscow).

# Structure of Talk


1. Definitions 23

2. Detailed Examples

   (a) Railway Computing Systems 25

   (b) Airline Business Computing 31

   (c) Air Traffic 37

   (d) Ministry of Finance 41

   (e) Manufacturing Systems 47

   (f) Enterprise Management 61

   (g) Multi-Script 71

   (h) DSS for SD 83

   (i) Federated GIS+DIS 85

   (j) LiMaCS 89

   (k) *&c.*

3. Research & Development 91

4. Why UNU/IIST's Interest? 105

5. Research Issues 107

6. Review & Conclusions 109

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

| |
|---|
| Foil 3 on projector 1. |

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

| |
|---|
| Another emphasis of the talk is to make precise the concept of infrastructure. Current definitions emphasize either socio-economical aspects or computing implementational facets. Our view is that an infra-structure is characterized as the syntactics set of two or more distinct sub-languages and similarly distinct, but significantly overlapping semantic universes of discourse. |

# Definitions

1. **World Bank: Infrastructure**

   *an 'umbrella' term for activities referred to as "social overhead capital";*

   *encompasses activities that share technical and economic features*

   *(such as economies of scale and spill-overs from users to non-users).*

2. **Technical: Infrastructures**

   *composition of parallel and communicating processes*

   *whose behaviour allows orderly*

   *(timely, in-corrupted and economical)*

   *interaction between processes, incl. people.*

   *(i.e. synchronization and communication of messages, materials, etc.)*

3. **UNU/IIST's Approach:**

   *enable designs that allow monitoring & controlling (ie. managing) any infrastructure like a business,*

   *and laying bare as many features of the infrastructure as technically possible*

   *so as to give users and other stake-holders a strong voice and real responsibility.*

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

First I show only foil 4. I will first show that overhead foil without coloured overlays. Then I will show it with one overlay showing a blue rectangle, and finally one with both one blue and several green rectangles.

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

As an example: the railway infrastructure covers a wide variety of sub-languages: that spoken by long range planners, that spoken by day-to-day planners; those spoken by operations staff — for example by ticket reservation agents, by station, line, cabin and locomotive staff, etc,

The blue overlay rectangle that I now show shall represent the base language that is common to all sub-languages. And the green rectangles that I show next shall illustrate each of the special, professional sub-languages.

# Detailed Examples

## RaCoSy: Railway Computing Systems

1. **Railway Strategic Planning & Devt.**

   Resource & Service Up/Downgrade:

   New Lines, New Stations, New Trains, New Services, ...

   Related Finance and Staff Assessments, Investments

2. **Tactics: Resource Scheduling & Allocation: S&A**

   Timetabling, Rolling Stock S&A, Staff S&A, ...

3. **Operations: Resource Deployment**

   Clients: Ticketing, Freighting, ...

   Traffic: Train Dispatching, Monitoring & Control

   Services: Information, ...

   Maintenance: Preventive, Adaptive, Corrective

4. **Statistics**

   Traffic, Usage, Wear & Tear, ...

5. **Staff Development**

   Awareness, Training, Testing, ...

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

| |
|---|
| Foil 4 on projector 1, foil 5 on projector 2. |

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

| |
|---|
| We shall not explain the formulas shown now.<br>But we shall claim that with a model as sketched — perhaps 2–3 more pages — we can cover most aspects of several railway system sub-languages: strategic and tactical planning, train dispatch personnel, etc. |

## Formal Domain Model 1: RaCoSy

**type**
U, C, L, S, P, N, Q
$\Sigma = (C \times C)$-**set**, $\Omega = \Sigma$-**set**
$TT = Tn \xrightarrow{m} (S \xrightarrow{m} (T \times T))$
$RE = Tn \xrightarrow{m} ((S \times S) \xrightarrow{m} (N \xrightarrow{m} Q\text{-}\mathbf{set}))$
$TF = T \xrightarrow{\sim} (Tn \xrightarrow{m} U^*)$, ...
**value**
Invariants
inv_N $\rightarrow$ **Bool**, ...
inv_TT: TT $\rightarrow$ **Bool**, ...
inv_TT_N: TT×N $\rightarrow$ **Bool**, ...
inv_TF_TT_N: TF×TT×N $\rightarrow$ **Bool**

Observers
state: U $\rightarrow \Sigma$, ...
xNLs: N $\rightarrow$ L-**set**, xSPs: S $\rightarrow$ P-**set**, xNSs $\rightarrow$ S-**set**
xinLsT: S × P $\xrightarrow{\sim}$ L-**set**, xoutLsT: S × P $\xrightarrow{\sim}$ L-**set**,

Generators
Open_L: U×U×N $\xrightarrow{\sim}$ N,
Close_L: U×U×N $\xrightarrow{\sim}$ N, ...
Sched: TT × N $\xrightarrow{\sim}$ TF-**set**,
BestSched: TF-**set** × N $\xrightarrow{\sim}$ TF, ..

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

Foil 6 on projector 1.. First I will read the below text.

Then after referring to the framed box (opposite page) I will elaborate on the laws.

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

But there are more than one way of expressing the semantics of a sub-language. This less abstract, more concrete formalization focus just on the train dispatch sub-language.

This sub-language model can be shown to be a refinement of the former model.

A third emphasis of this talk is that of illustrating laws of man-made systems.
2mm] Kirschoff's law says OVER A SUITABLY CHOSEN TIME INTERVAL WE HAVE THAT THE NUMBER OF TRAINS ARRIVING AT A STATION, MINUS THE NUMBER OF TRAINS TAKEN OUT OF SERVICE (I.E. ENDING THEIR JOURNEY) AT THAT STATION, PLUS THE NUMBER OF TRAINS PUT INTO SERVICE (I.E. STARTING THEIR JOURNEY AT THAT STATION, EQUALS THE NUMBER OF TRAINS DEPARTING FROM THAT STATION.
You will be surprised to know how often programmers forget to have their station management software obey even this simple law!

To understand the "No Ghost Trains" law we must consider the intrinsic concept of traffic as a function which is total over the time interval say from the departure of a first train to the arrival of a last train. In that interval: if a train appears in the traffic at times $t_i$ and $t_j$ then it must appear continuously at any time in between. We will later revert to the concept of "ghost trains".

Finally "God does'nt play dice": that is, no pair of hands lift trains from lines or tracks and interchange them. Put it differently: two trains, A and B, progressing orderly down a line, A first, B following, can not suddenly, in one time instance, change relative position with B now being first!

The point about our formal definitions and the laws is this: the formal definitions must capture all these laws. They must be provable from the theories.

Did I mention that UNU/IIST has a large scale railway software development with the Chinese Ministry of Railways?

## Formal Domain Model 2: RaCoSy

**type**

N = (Ln $\overrightarrow{m}$ Line) × (Sta $\overrightarrow{m}$ Station)

Line = Len × ... × LnTyp
LnTyp == up | down | updown

Station = (Tr $\overrightarrow{m}$ Track)
Track = Len × ilns:L-**set** × olns:L-**set** × TrTyp
TrTyp == line|siding|platform|freight|...

TT = Tn $\overrightarrow{m}$ StaVis* × × Time Sta
StaVis = (arr:Time × Sta × Tr × dep:Time × Ln)

Alternative Model (2) is a Refinement of Model 1!

**Laws:**

- "Kirschoff's" Law
- "No Ghost Trains"
- "God Does'nt Play Dice"
- *&c.*

General Transportation Theory

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

| |
|---|
| Foil 7 on projector 1. |

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

| |
|---|
| Although our next example is also a transportation system we focus here on the sub-language spoken by airline business planners at the strategic, the tactical, and as we shall see on the next overhead foil ... |

# ABC'2000: Airline Business Computing

In many respects re-usable concepts of RaCoSy:

1. Strategic Planning, Policies:
   (a) Route Networking:
   Feeder, Regional, Global Carrier
   (b) Resource Build-up/Down-size:
   Monies, Staff, Aircrafts, Material, Facilities
   (c) Financial Plan, Pricing, ...
   (d) *&c.*

2. Tactical Planning:
   (a) Resource Scheduling & Allocation
   Timetables, Maintenance, ...
   Material & Staff Schedules, ...
   (b) Cash Flow Tuning,
   Airline↔Bank Liquidity Reconciliation, ...
   (c) Labour Market Relations
   (d) Advertising, Customer Programmes
   (e) *&c.*

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

Foil 7 on projector 1, foil 8 on projector 2.

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

... operational planning level. We did not illustrate those management sub-languages for the first case of Railways.

Now let us look at what laws may be of interest:

The *Law of Low Resource Bound* expresses the following: For every aircraft type we know how many passenger and freight–ton kilometers it can service over a year. If at the same time we now which expected number of passengers and freight tons over which routes, then we can compute the lowest number of aircrafts needed in order to meet expectations.

The *Law of High Resource Bound*, also known as the *Law of Diminishing Returns* stipulate that increasing the number of aircrafts over the low resource bound will increasingly lead to diminishing profits, eventually to losses!

The *Law of Passenger & Freight Service Quality Enhancement*, stipulates that service improvements above, or service curtailments below certain thresholds will likewise lead to diminishing profits, eventually losses.

The ABC'2000 project is together with Vietnam Airlines!

## ABC'2000 — continued

3. Operational Deployment

(a) Maintenance Engineering & Fueling
(b) Ticketing/Reservations: Passenger & Freight
(c) Passenger Check-in, Freight Loading/Unloading
(d) *&c.*

(e) Interface to Gate Control/Boarding
(f) Interface to Traffic Control, Dispatch & Arrival
(g) *&c.*

**Laws:**

- Law of Low Resource Bound:
  Min.# of Aircrafts to Satisfy Forecast
  Passenger Km. + Freight Km.
- Law of High Resource Bound:
  Law of Diminishing Returns
- Law of Product Quality Enhancement:
  Law of Efficient Resource Handling
- *&c.*

See also: **Enterprise Management**

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

| |
|---|
| Foil 9 on projector 1, foil 5 on projector 2. |

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

| |
|---|
| Again we must make sure that any model of strategic and tactical planning as well as operational deployment, satisfies these and other laws. |

## ABC'2000

Strategy

**type**

RES = Arc-**set** × Staff-**set** × Cap

NW = A $\overrightarrow{m}$ A-**set**

ETF = A $\overrightarrow{m}$ (A $\overrightarrow{m}$ ((T × T) $\overrightarrow{m}$ **Nat**))

Cap = **Nat**

Arc, Staff, A, T

**value**

strategy: NW × ETF × Cap $\xrightarrow{\sim}$ (RES $\xrightarrow{\sim}$ (RES × Cap))

...

Tactics

**type**

cTT = T $\overrightarrow{m}$ (A $\overrightarrow{m}$ (A $\overrightarrow{m}$ ((T × T) × Arc)))

rTT = F $\overrightarrow{m}$ (A $\overrightarrow{m}$ (A × (T × T))), F

**value**

titbl: ETF × RES $\xrightarrow{\sim}$ cTT-**set**

bsttitbl: cTT-**set** × Cap $\xrightarrow{\sim}$ rTT

...

Deployment

**type**

BK = F $\overrightarrow{m}$ ((A × A) $\overrightarrow{m}$ (P|C) $\overrightarrow{m}$ Q-**set**), ...

**value**

res: (P|C) × Q-**set** × BK $\xrightarrow{\sim}$ BK, ...

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

Foil 10 on projector 1.

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

The sub-languages of air traffic include those of the airport ground staff, that is those people (i) who work out there on the tarmac helping to taxi aircrafts to and from their gate positions, (ii) the pilot briefing room staff, (iii) the dispatchers, (iv) the air traffic controllers, (v) the pilots, radio-engineers, etc.

Observable disruptions can be properly defined and are the basis for re-schedulings. This requires proper definitions of scheduled and real traffic. Subsequent requirements capture may then define exactly what kind of computer support such traffic monitoring & control shall have.

As for the railway transportation system we expect *Kirschoff's Law*, the *Law of no "ghost flights"* to apply. Whereas a train may stand still on a line for any amount of time, an aircraft must continuously move while in the air: hence *The Law of Monotonicity.*

The ATC'2000 project is starting up this summer with ASC: Asia Simulation & Control of China and the CAAC, Civil Aviation Authority of China. AeroThai of Thailand may join the project.

# ATC'2000: Air Traffic

Well-defined Interfaces to Airline Businesses. Plus:

1. Airspace

   Airports:

   Tarmac, Briefing Room, Gate Control, &c.,

   Airdomes, Air Lanes/Corridors, Networks

2. Timetables

   Airline, Airport (Gate), and Network Schedules

3. Traffic

   Scheduled vs. Real Traffic

   Scheduling and Re-scheduling

4. Air Traffic Control

   Monitoring & "Control":

   Dispatch, Arrival, En-route

**Laws:**

- Monotonicity
- *&c.*

See also: **Railway Systems**

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

| |
|---|
| Foil 10 on projector 1, foil 11 on projector 2. |

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

| |
|---|
| And we must demand that these laws be properties of our formal model.<br><br>The model shown captures the topological nature of Airdomes (D), Airways or Corridors (W), and the position of aircrafts (E, for envelope). Thus the model of the real domain of of traffic relies very much on a good understanding of topology. Its realization correspondingly implies massive use of parallel computing over 3 dimensional spaces to help predict conflicting aircraft journies to avoid crashes, yes to avoid even "near-misses". |

**type**
  AS = inas:A-**set**
    × (A $\overrightarrow{m}$ (D × (A $\overrightarrow{m}$ W-**set**)))
    × ouas:A-**set**
  TT = F $\overrightarrow{m}$ (A $\overrightarrow{m}$ (T × T))
  sTF = T $\xrightarrow{\sim}$ (F $\overrightarrow{m}$ E)
  rTF = T $\xrightarrow{\sim}$ (F $\overrightarrow{m}$ E)
  D, W, E
**value**
  Invariants
  inv_AS: AS → **Bool**
  inv_TT: TT → **Bool**
  inv_TF: TF → **Bool**
  inv_TT_AS: TT×AS → **Bool**
  inv_TF_TT: TF×TT → **Bool**
  inv_TT_TT_AS: TT×TF×AS → **Bool**
  Observers
  disruption: rTF×sTF×((rTF×sTF)→**Bool**)-**set**→T
  Generators
  Open_A: A×AS $\xrightarrow{\sim}$ AS, Close_A: A×AS $\xrightarrow{\sim}$ AS
  Open_W: W×(A×A)×AS $\xrightarrow{\sim}$ AS
  Close_W: W×(A×A)×AS $\xrightarrow{\sim}$ AS
  Schedule: AS×TT $\xrightarrow{\sim}$ sTF-**set**

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

Foil 12 on projector 1.

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

An entirely different kind of infrastructure, one that is almost wholly dependent and focused on people is that of a ministry of finance. Sub-languages may be related to taxation systems, budgetary systems, expenditure systems, external debt management, or other.

The first law concerns sound fiscal policy: the value of increased, or decreased production stands in a direct relationship to the bank notes (and coins) in circulation.

The second law concerns audit-ability: every transaction must be properly recorded.

This project is paid for by the Vietnam Ministry of Finance. Currently we have five Fellows at UNU/IIST from Hanoi.

# MoFIT: Ministry of Finance IT

1. Budgetary System
   (a) Accounts: Credits, Debits, Balance, Statements, ...
   (b) Transactions: Receipts, Payments, Transfers: In/Out, ...
   (c) Budgets: Simple and Compound: Hierarchical & Distributed
   (d) Policies: Rules & Regulations: Enforceable, Reportable,
2. Audit & Security
3. Taxation System [Tax, Duty, Fee, *&c.*] Collection
4. Expenditures: Funds Disbursement/Transfer
5. Debt Management

   *&c.*

**Laws:**

- "Money+Production" in = "Money+Production" out!
  The Law of Responsible Note Printing Press
- Law of Honesty: Audit-ability
- *&c.*

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

| |
|---|
| Foil 12 on projector 1, foil 13 on projector 2. |

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

| |
|---|
| These laws must naturally be obeyed by any model we choose to write down. |

## MoFIT — continued

**type**
  Acct, Staff, Data, Time
  Cmd = Paym | Rcpt |Bal | In | Out | Stmt | ...
  Sum = **Int**
**value**
  credit: Acct × Sum × Staff → Acct
  debit: Acct × Sum × Staff → Acct
  balnc: Acct × Staff → Sum
  xfer: Acct × Acct × Date × Sum × Staff
    → (Acct × Acct)
  stmt: Acct × Staff → (Cmd × Sum)*

**type**
  Bn
  Cmd′ = Cmd | Accts | ...
  Budget = Bn $\xrightarrow[m]{}$ (Acct | Budget)
**value**
  accts: Budget × Staff → Acct-**set**
  MoF: Budget

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

| |
|---|
| Foil 14 on projector 1, foil 13 on projector 2. |

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

| |
|---|
| Ministry rules & regulations must be reflected in suitable predicates and functions of the model, and, most likely be required implemented in appropriate software support of ministry computing. |

## MoFIT — Cont'd.

**type**
  Rn
  Cmd″ = Cmd′ | Rept
  Rept = Rn
  Policies = Enforceable | Reportable
  Enforceable = (MoF → **Bool**)-**set**
  Reportable = Rn $\xrightarrow[m]{}$ (MoF $\xrightarrow{\sim}$ Report)
  Report = (Staff × Date × Time × Cmd″)*
**value**
  Invariant
  inv_MoF: MoF × Enforceable → **Bool**
  inv_MoF(mof,enf) ≡
    ∀ p:Enforceable · p ∈ enf ⇒ p(mof)
  Generators
  grept: MoF × Reportable $\xrightarrow{\sim}$ Report-**set**
  grept(mof,rpt) ≡ { r(mof) | r:Reportable · r ∈ **rng** rpt }

  srept: MoF × Reportable × Rn $\xrightarrow{\sim}$ Report
  srept(mof,rpt,rn) ≡ (rpt(rn))(mof)
  **pre**: rn ∈ **dom** rept

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

| |
|---|
| Foil 15 on projector 1. |

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

Next we tackle a rather more complicated infrastructure: that of the collection of sub-language of the many enterprises belonging to an industry — its market so-to-speak.

On this overhead foil we show the increasingly detailed facets of the manufacturing sub-language: from the market: that is, consumers, suppliers, traders and producers, via the individual enterprises and their marketing, finance, administration and production, to the product transformation process and the underlying production system. Again we must model the terms of these sub-languages.

This project has brought two Fellows from The Philippines to UNU/IIST for nine months.

# MI$^2$CI: Manufacturing Systems

## Mfg. Industry Inform. & Cmd. Infrastructures

1. The Market

   Consumers, Suppliers, Traders, Producers

2. The Enterprise

   Marketing, Finance, Administration, Production

3. Transformation Process

   Order handling, Scheduling, Work-order processing, Material processing, Product processing

4. Production System

   Product cycle, Process planning & improvement, Plant location, Plant layout, Material handling, Quality standards & control, Inventory control, Plant engineering, Maintenance

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

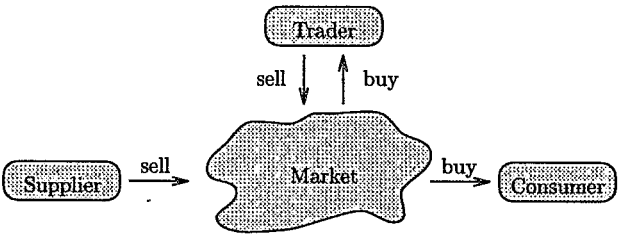
| |
|---|
| Foil 15 on projector 1, foil 16 on projector 2. |

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

| |
|---|
| The Market speaks about supply and demand. The consumer expresses demand, and buys and consumes products. |

# MI²CI: Market, Consumer, Supplier

## Single Product




**type**
  Market = **Int**
**value**
  more_supply: **Nat** × Market → Market
  more_demand: **Nat** × Market → Market

**type**
  Consumer
**value**
  demand: Consumer × **Nat** → Consumer
  buy: Consumer × Market → Consumer × Market
  consume: Consumer × **Nat** ~→ Consumer

# Notes for Interpreters, Projectionist and Speaker

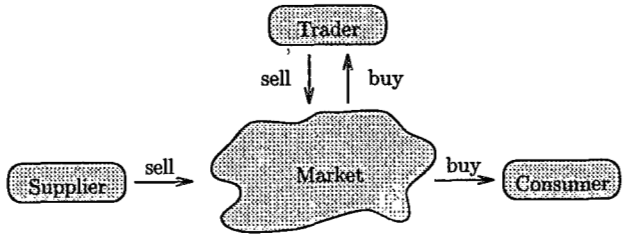
**Notes to Interpreters :**

| |
|---|
| Foil 17 on projector 1, foil 16 on projector 2. |

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

| |
|---|
| Suppliers supply and sell products; and traders combine the functions of consumers and suppliers. |

## MI²CI: Single Product — continued




**type**
  Supplier = **Nat**
**value**
  supply: Supplier × **Nat** → Supplier
  sell: Supplier × **Nat** × Market ~→ Supplier × Market

**type**
  Trader = Consumer × Supplier
**value**
  demand: Trader × **Nat** → Trader
  buy: Trader × Market → Trader × Market
  sell: Trader × Market × **Nat** → Trader × Market

# Notes for Interpreters, Projectionist and Speaker

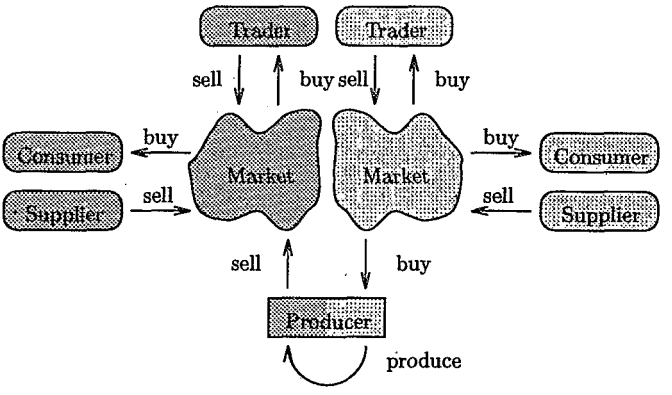
**Notes to Interpreters :**

Foil 17 on projector 1, foil 18 on projector 2.

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

A market whose products contain parts supplied from elsewhere implies that there are producers and that there are different markets.

## MI²CI: Multiple Product




**type**

Pn

Market = Pn $\xrightarrow[m]{}$ **Nat**

Consumer :: prod:Pn dem:**Nat** stock:**Nat**

Supplier :: prod:Pn stock:**Nat**

Producer = Consumer × Supplier

**value**

more_supply: Pn × **Nat** × Market → Market

more_demand: Pn × **Nat** × Market → Market

demand: Producer × **Nat** → Producer

buy: Producer × Market $\xrightarrow{\sim}$ Producer × Market

sell: Producer × **Nat** × Market $\xrightarrow{\sim}$ Producer

produce: Producer × **Nat** $\xrightarrow{\sim}$ Producer

Trader unchanged

# Notes for Interpreters, Projectionist and Speaker

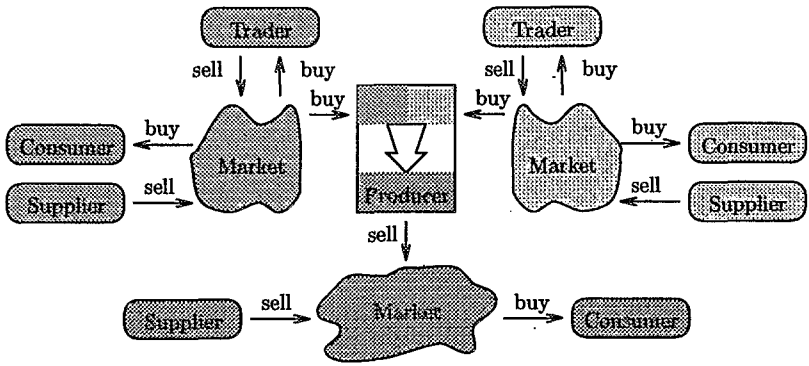
**Notes to Interpreters :**

| |
|---|
| Foil 19 on projector 1, foil 18 on projector 2. |

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

| |
|---|
| In general there will be any number of markets. |
| Now which laws are of interest to us, that is: which properties must the models possess? |
| Transitivity: Part A may be composed from part B, and B from C; if so, then part A is composed from C. But part D cannot be composed from itself. |
| Scavenging: An automobile put together from wheels and tires, body, etc., cannot give rise to wheel, tire and body products — although that is what you may indeed observe when someone illegally "scavenges" somebody else's car! |
| Etcetera. |

# MI²CI: Multiple Sub-products




**type**

Producer :: sup: Supplier
con:(Pn $\xrightarrow{m}$ Consumer)
bill:(Pn $\xrightarrow{m}$ **Nat**)

**Laws:**

- Law of Non-cyclic Supply Chain

  Transitivity but not Reflexivity of Transformation
- Laws against "Scavenging"
- Law of Supply & Demand
- Demand imply Consumers
  Suppliers imply Producers
- Producers imply Multiple Products
  Multiple Products imply Markets
- *&c.*

# Notes for Interpreters, Projectionist and Speaker

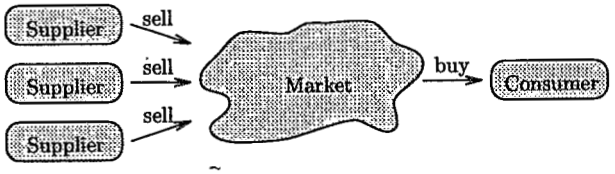
**Notes to Interpreters :**

| |
|---|
| Foil 20 on projector 1, foil 19 on projector 2. |

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

| |
|---|
| We can certainly also deal with fair market sharing — something the US believes it need tell Japan and China about! |

## MI²CI: Marketing — Fair Share




**type**
  Sn, Pn
  Market :: tot:**Int**
    dis:(Sn $\overrightarrow{m}$ **Nat**)
**value**
  more_supply: Sn × **Nat** × Market → Market
  more_demand: **Nat** × Market → Market

# Notes for Interpreters, Projectionist and Speaker

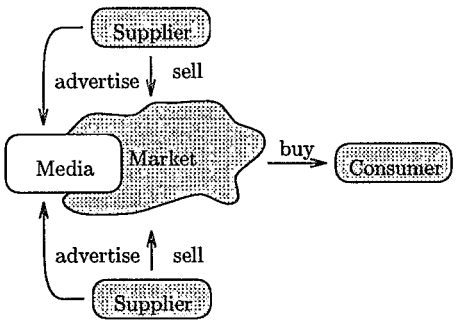
**Notes to Interpreters :**

| Foil 20 on projector 1, foil 21 on projector 2. |
|---|

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

| And we can formalize crucial advertising sub-languages. |
|---|

## MI²CI: Advertising




**type**
  Sn, Pn
  Media = Sn $\xrightarrow{m}$ **Nat**
**value**
  advertise: Sn × **Nat** × Media → Media
  adv_value: Sn × Media → **Nat**
  adv_prop: Sn × Media → **Real**
**type**
  Market :: dis:(Sn $\xrightarrow{m}$ **Nat**)
            tot:**Int**
**value**
  more_demand: **Nat** × Media × Market → Market
**type**
  Supplier :: prod:Pn
              stock:**Nat**

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

Foil 22 on projector 1.

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

Again we change subject: now to that of the sub-language shared by all top- and middle-level managers: strategists and planners of enterprises. We briefly covered such a sub-language when we covered the airline business example. Now we abstract the sub-language to cover several kinds of enterprises' resource acquisition, disposal, scheduling and allocation concerns.

Before we construct our models, and as we should have done in all other, previous and following cases, we enunciate, we express, the laws.

Typical we find that these laws are today well covered by the operations research and business management disciplines: Rate Monotonic Scheduling is one such law. There are many laws.

# Enterprise Management

## Strategic, Tactical & Operational Planning

1. Resources:
   (a) Spatial Allocation
   (b) Time & Schedules
2. Clients; Products & Tasks
3. Orders; Objectives, Estimates, Performance
4. Planning
   (a) Strategic
   (b) Tactical
   (c) Operational
5. Operations

**Laws:**

- Rate Monotonic Scheduling
- *&c.*

Adherence to Operations Research Laws

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

Foil 22 on projector 1, foil 23 on projector 2.

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

In our model we focus on mobile resources: their locations, their being generally available (i.e. scheduled) or not available in certain time intervals, and their being both allocated and scheduled.

## Strategic, Tactical & Operational Planning — continued

**type**

R, VAL, On, RTyp
RS = R-**set**
O = VAL × R* $\xrightarrow{\sim}$ VAL
OS = On $\overrightarrow{m}$ O

L
LQ = L | L-**set** | L-**infset** | L−lsit | L-**infset**
GRA = R $\overrightarrow{m}$ LQ

T
TS = T-**set**
TI = {| (t,t') | t,t':T • t<t' |}

GRS = TI $\overrightarrow{m}$ RS
GRAS = TI $\overrightarrow{m}$ (RS $\overrightarrow{m}$ LQ)

**value**

allocate, de_allocate
schedule, de_schedule
alloc_sched, de_alloc_sched

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

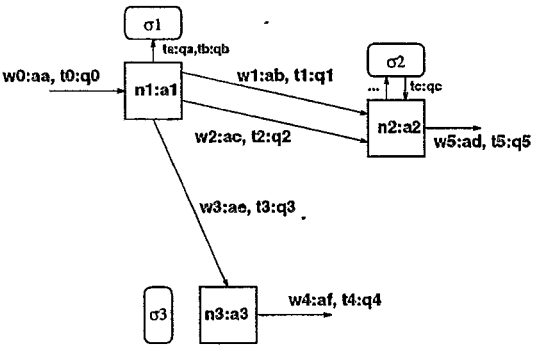
| |
|---|
| Foil 24 on projector 1, foil 23 on projector 2. |

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

| |
|---|
| Allocation was used above in the straightforward sense of binding to physical, geographical locations. Now we take the more narrow view of allocation as binding resources to tasks, for example the production task of a manufacturing enterprise or the task of offering specific air travel to specific passengers. We can describe such production tasks in terms of for example graphs: Nodes denotes product transformation actions other than flow, edges denote flow of products — as on the production floor of a manufacturing plant. Thus nodes and edges must be bound to action and flow (transport) resources and time intervals. |

# Enterprise Management — continued

## Products & Tasks: A Production Graph




**type**

P, Pn, N, W

PD = inputs:(Nxt × Loc)
× next:(N $\xrightarrow{m}$ (Nxt × Stg))
× outputs:(Next × Loc)
× Acts

Nxt = N $\xrightarrow{m}$ (W $\xrightarrow{m}$ Loc)

Loc = RTyp $\xrightarrow{m}$ **Nat**

Stg = RTyp $\xrightarrow{m}$ **Nat**

ASPD = PD × NWAS

NWAS = (N|W) $\xrightarrow{m}$ GRAS

ASPDS = Pn $\xrightarrow{m}$ ASPD

## Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

| |
|---|
| Foil 25 on projector 1, foil 24 on projector 2. |

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

| |
|---|
| Now we can speak about production tasks, clients, their orders, and past and future performance. The latter amount to statistics, respectively estimates concerning production (sales, etc.). |

## Enterprise Management — continued

### Tasks

**type**
  Task = ASPD × Trace
  Trace = T $\xrightarrow{\sim}$ (N|W)-**set**
  Tasks = T $\xrightarrow{\sim}$ (Pn $\xrightarrow{m}$ Task-**set**)

### Clients, Orders, Performance

**type**
  C, On
  O = Pn $\xrightarrow{m}$ (TI $\xrightarrow{m}$ **Nat**)
  CO = C × O
  OS = C $\xrightarrow{m}$ (On $\xrightarrow{m}$ O)

  BO = R $\xrightarrow{m}$ **Int**
  S,E = TI $\xrightarrow{m}$ (LQ $\xrightarrow{m}$ BO)
  SE = S × E
**value**
  gather: (T $\xrightarrow{\sim}$ ASPDS) × Tasks $\xrightarrow{\sim}$ S
  project: S × ... $\xrightarrow{\sim}$ E

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

Foil 25 on projector 1, foil 26 on projector 2.

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

Finally we can summarize — and later further detail — the strategic, tactical and operational planning activities. Thus we can characterize strategic management as one that is concerned with decisions as whether to acquire new, and/or dispose of old resources, and when. The **sp** function denotes this. However, such decisions are only temporary and must be negotiated with the market. An airline may, as a result of increased passenger estimates on certain destinations require access to shared runways. But other enterprises are competing for the same resources. Negotiation results in choosing one of several feasible resource plans. The function **neg** gives the signature of such a function. Now estimates may have to be revised before we can proceed via tactical planning (shown as function **tp**) — which further refines the resource locations and schedules — to operational planning (shown as function **opl**) — which binds tasks to resources. Finally we deploy all resources, ie. enter into action, that is: into operations (shown as function **ops**).

What we have done here is just to give type names to resources, locations, time intervals and their individual or combined composition into allocations and schedules, and to give just the signature of the various planning functions. Already this clarification is, we claim, tremendously beneficial, and is something we from computing science seems better at than people from operations research, industrial engineering, business management, etc.

Whether these functions are actually computable is another matter. To the extent they are we may wish, later, in a subsequent Requirements Capturing phase, to specify what needs being computed. To the extent they are not computable we may at least be able to record all the resources, their instantaneous values, locations and schedules and their compositions into allocations and schedules so that humanly performed planning (decision support) is well-supported.

## Strategic, Tactical & Operational Planning; Operations

**type**
  GRS = TI $\overrightarrow{m}$ RS
**value**
  [strategic planning]
  sp: RS × SE × TI $\xrightarrow{\sim}$ GRS-**set**

**type**
  B
**value**
  [external resource negotiation]
  neg: (B × GRS-**set**) × (B $\overrightarrow{m}$ GRS-**set**) $\xrightarrow{\sim}$ GRS
  [resource resolution]
  rsl: E × GRS-**set** × GRS $\xrightarrow{\sim}$ GRS
  [revision **of** estimate]
  ree: E × GRS $\xrightarrow{\sim}$ E
  [tactical planning]
  tp: E × OS × GRS $\xrightarrow{\sim}$ GRAS
  [operational planning]
  opl: GRAS × OS $\xrightarrow{\sim}$ ASPDS
  ops: ASPDS $\xrightarrow{\sim}$ Tasks

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

Foil 27 on projector 1.

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

A next example is quite surprising. It concerns the universe of discourse implied by the sub-languages spoken by people involved in the document creation, editing, formatting, printing & display, and publication trades.

We are especially concerned with such documents which contain, almost arbitrarily composed, sentences of text from up to four different classes of scripts — where each class is characterized by its own natural orientation and direction. Mongolian texts, for example, are ordered in top-to-bottom read columns appearing left to right; classical Japanese columnized texts are also read top-to-bottom, but ordered right-to-left. English, as you see it on my overhead foil, is read left-to-right in rows ordered top-to-bottom, while for example Arabic is read right-to-left.

Now what laws could possibly determine the semantics of such texts when viewed as documents subject to creation, , editing, formatting, printing & display, and publication?

Perhaps surprisingly, we suggest that whatever function we perform on a document, its "entropy" cannot change. This may mean that the functions we deal with apply to pairs of documents: the "real" document being operated upon, and the "virtual" **delta change** document with which the operator (the writer, the editor, the typographer (i.e. the formatter), etc.) is manipulating the document.

Another law also seems surprising, perhaps not when viewed only in an editing context, but seemingly when viewed in a formatting context: namely that each editing and each formatting operation has a "natural" inverse. The **editing undo** button for editing need its complement in an **unformat button**!

The project is joint with Mongolia!

# Multi-Script

## Multi-lingual Script System

1. All Scripts
2. Four Directions
   Mongol: TD/LtR; Chinese: TD/RtL;
   English: LtR/TD; Arabic: RtL/TD
3. Any combination
4. Any Smallest part ("Frame")
5. Font: UNICODE Standard
6. Text Format:
   New Document Processor Standard

**Laws:**

- Entropy: Constancy
- Reversibility:
  Inverse of Editing & Formatting Functions
- *&c.*

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

| Foil 27 on projector 1, foil 28 on projector 2. |
|---|

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

| So we need properly model the sub-language of core documents, ... |
|---|

## Multi-Script — continued

### Core Model

1. Documents as Frames
2. Frame "text" having Orientation, Metrics, and a list of Strings
3. Strings being lists of Tokens
4. Tokens being either Characters or Frames

### Core Model

**type**

Document = Frame
Frame = Direction × Format_Info × String*
String = Token*
Token = Character | Frame
Direction = Entry_Pt × Orientation × Rotation
Entry_pt == left | right
Orientation == horiz | vert
Rotation = {| r:**Nat** · r ≤ 3 |}
Format_Info = Font_Nm × Font_Sz × TypeFa × Col × ...
Character

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

Foil 29 on projector 1, foil 28 on projector 2.

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

...and essential functions over (operations upon) core documents, ...

## Multi-Script — continued

### Core Model — continued

**type**
  Index = **Nat**$^*$

**value**
  incr_lst: Index → Index
  remv_lst: Index → Index
  valid_idx: (Document|Frame|Token) × Index → **Bool**
  nxt_idx: Document × Index → Index
  unstack: Document × Index → Index

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

| |
|---|
| Foil 34 on projector 1, foil 30 on projector 2. |

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

| |
|---|
| Documents come in many forms: books, newspapers, administrative forms, letters, etc. |

## Multi-Script — continued

### Document Models

1. *Book*
   (a) Books being lists of Chapters
   (b) Chapter having a Display Line, Text and a list of
2. *Newspaper:*
   (a) A Newspaper as a list of Pages
   (b) Pages with Heads, Foots and Column Blocks
   (c) Single/Multiple Column Blocks
   (d) S/M Column Texts, Photos, ...
   (e) Inserts
   (f) *&c.*
3. *Administrative Form:*
   (a) Head
   (b) Fields: Title, Fill-in Text, Help Text
   (c) Foot
4. *Letter*
5. *&c.*

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

| |
|---|
| Foil 31 on projector 1, foil 30 on projector 2. |

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

| |
|---|
| ... with these linguistic terms having precise meanings with precise relations determining when a document can be formatted as a book, or as a newspaper, etc. |

## Multi-Script — continued

### Document Model

**type**

Book = Chap*
Chap = Displ × Txt (Sect × **Text**)*
Sect = Displ × Txt (Subs × **Text**)*
SubS = ...

BookForm
NewsForm

**value**

inv_Book: D → **Bool**
Doc_Book: D ~→ Book
Book_Doc: Book → D
FormatBk: Book × BookForm ~→ Print

inv_News: D → **Bool**
Doc_News: D ~→ News
News_Doc: News → D
FormatNw: News × NewsForm ~→ Print

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

Foil 31 on projector 1, foil 32 on projector 2.

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

Finally actual formatting functions can be dealt with as well as functions that retrieve core documents from books, newspapers, etc.

In this example we show how a new kind of research study which takes place in a world where computing is possible, can dramatically change the "business" of document production.

If you thought TEX and LATEX were the end-of-the-road, please reconsider!

# Multi-Script — continued

## Format Information

1. Page size
2. Text Width, Height
3. Margin Parameters
4. *&c.*

## Transformation Functions

1. Core document $\leftrightarrow$ Book
2. Core document $\leftrightarrow$ Newspaper
3. Core document $\leftrightarrow$ Form
4. Core document $\leftrightarrow$ Letter

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

| |
|---|
| Foil 33 on projector 1. |

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

| |
|---|
| Development deals with changing resources. Sustainable development must make sure that certain equities are maintained within reasonable bounds (here expressed as Fuzzy measures).<br><br>This example shows that we can indeed apply our domain analysis techniques. It also suggests a **Law of Decreasing Sustainability:** namely that development invariably leads to less future options.<br><br>This and the next example illustrates a concern of the UN System: Agenda'21. It seems that The World Bank may fund a large project for constructing a Decision Support System Workbench. UNU/IIST is already collaborating with research institutes in Asia (China, Hong Kong, Singapore, India) and Africa (Egypt, Zaire, Cameroun, etc.) on the DSS for SD ideas. |

# Decision Support for Sustainable Development

## Resources, Attributes, Values, Indicators, Equities

**type**
  R
**value**
  D′: $R^* \xrightarrow{\sim} R^*$
**type** A, I, Fuzzy
  RAIS = R $\xrightarrow[m]{}$ (A $\xrightarrow[m]{}$ **I-set**)
  E = (RAIS × RAIS) $\xrightarrow{\sim}$ Fuzzy
  ES = En $\xrightarrow[m]{}$ E
  D = RAIS $\xrightarrow{\sim}$ RAIS
  DEV = RAIS × ES × D
**axiom**
  ∀ (rs,es,d):DEV · ∀ e:E · e ∈ **rng** es ⇒
    ∃ f:Fuzzy · e(rs,d(rs)) ≤ f

**Laws:**

- Law of Decreasing Sustainability:
  *The Almighty & and his People!*
- *&c.*

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

First I show overhead foil page 34. During the reading of the below text I also add, on the other projector, overhead foil page 35.

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

Development usually requires planning and planning usually entails access to geo- & demographic data. As yet another infrastructure we here model the terms of languages and their shared universes of discourse, namely those spoken and those data manipulated by geographers, demographers, planners, etc. In reality they refer to a globally distributed set of GIS-s and Dis-s.

Here we model that distribution. We also model operations on GIS and DIS data, even combinations of such data and repositories for the results of such operations. From that to actually constructing a federated system for supporting this universe of discourse with computing & communication there is only a short distance, intellectually, but a big one technologically!

Whether we just speak about the reality of all the different information repositories or their software realization, certain laws must be satisfied: we cannot improve the value of the data input by manipulating it, we can only make that data more "accessible"!

# Federated GIS+DIS: GaDI$^2$S

- Global & Distributed, Geo- & Demographic Information Infrastructure Systems
- Software Technology for Decision Support Systems for Sustainable Development

**type**

$\Sigma$ = GaDIIS × DataBase × STG

GaDIIS = (D $\overrightarrow{m}$ (T $\overrightarrow{m}$ ((V $\overrightarrow{m}$ (DRef × GaDIIS))
× (Uop $\overrightarrow{m}$(T × uop:(Data$\tilde{\rightarrow}$Res)))
× (Nop $\overrightarrow{m}$ ((T* × T)
× (Data* $\tilde{\rightarrow}$ Res))))))

STG = (D × T × V)* $\overrightarrow{m}$ Res

DRef = Loc × (Data $\tilde{\rightarrow}$ Data)

DataBase = Loc $\overrightarrow{m}$ Data

**Laws:**

- "Garbage In / Garbage Out":
  No Miracle Information Creation
- *&c.*

# Notes for Interpreters, Projectionists and Speaker

**Notes to Interpreters :**

| |
|---|
| Overhead foil page 35 is put on the second projector while text for page 34 is read. |

## GaDI$^2$S Terms

- $\Sigma$: System State
- D: Information Names
- T: Types
- V: Versions
- Uop: Unary Operators
  on Geo- or Demographic, etc. Data
- res: Result Type
- uop: Unary Operation
- Res: Result value
- Nop: N-ary Operators
  on Geo- or Demographic, etc. Data
- nop: N-ary Operation
- Stg: "scratch" Storage
- Loc: Location
- dgf: Data (incl. map) Generalization Function

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

| |
|---|
| Foil 36 on projector 1. |

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

Our last example is that of the real world of libraries, publishers (as seen in the context of libraries) and borrowers (again as seen in the context of libraries). Traders are the devices by means of which librarians are to know about the existence of publishers and borrowers, and publishers and borrowers are to know about libraries. In this world we must also model the operations of librarians, borrowers, etc.

I leave it to you to think up some laws!

Currently UNU/IIST is planning this Open Distributed Processing oriented implementation project with University of Macau.

# LIMaCS

**Library Information Monitoring & Command Systems**

**type**

L, P, Borrowers, D, ...Decsr..., B, Date

$\Sigma = (L \xrightarrow{m} LIB) \times (P \xrightarrow{m} PUB) \times Trader \times Borrowers$

$LIB = ACQ \times CTLG \times CLST \times BORR$

$PUB = D \xrightarrow{m} DocDescr$

$Trader = (L \xrightarrow{m} LibDescr) \times (P \xrightarrow{m} PublDescr) \times ...$

$ACQ = D \xrightarrow{m} DocDescr'$

$CTLG = D \xrightarrow{m} (DocDescr'' \times tot{:}\mathbf{Nat} \times lent{:}\mathbf{Nat})$

$CLST = D \xrightarrow{m} B^*$

$BORR = B \xrightarrow{m} (D \xrightarrow{m} Date)$

**value**

ordr: $L \times D \times P \times \Sigma \xrightarrow{\sim} \Sigma$

ctlg: $L \times D \times \Sigma \xrightarrow{\sim} \Sigma$

borw: $B \times L \times D \times \Sigma \xrightarrow{\sim} \Sigma$

**Laws:**

- Laws of Preservation
- *&c.*

# Notes for Interpreters, Projectionists and Speaker

**Notes to Interpreters :**

| |
|---|
| Foil 37 on projector 1. |
| Foil 37 is basically read as its text shows. |

# Research & Development

- Question:

  In view of recurrent and spectacular failures of large–scale software development projects:

  What justifies our 'tackling' the above projects?

- Answer:

  The use of :

  - **Abstraction**
  - **Step-wise Refinement**
  - **Design Calculi** (Formal Methods)
  - **Separation of Concerns:**
    * Domain Analysis
    * Requirements Capture
    * Software Development
  - **Laws of Man-made Systems**

- Laws & Laws:

  - A-priori Laws: Model Validation
  - A-posteriori Laws: Invariants: Model Verification

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

Foil 37 on projector 1, foil 38 on projector 2.

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

With respect to the Separation of Concerns we emphasize the decomposition into programming and Software Engineering — between the Design Calculus and Scientifically driven parts and the Technology and Engineering driven parts.

At UNU/IIST we practice this overall development approach and find rich opportunities to further research their foundations.

# The Methodology: PM+SE

— in the "Large" & in the "Small"

Programming Methodology **PM**

1. Domain Analysis 40, D/A
2. Requirements Capture 41, R/C
3. Software Architecture 42, S/A
4. Software Design & Coding

Software Engineering **SE**

5. Version Ctrl. & Config. Mgt.
6. Test case Gen. & Validation
7. Hyper-Text Documentation

   &c.

**PM+SE**

8. — Supported by Mathematics-based CASE Tools

   Comprehensive: Syntactic, Semantic & Pragmatic

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

Foil 39 on projector 1, foil 38 on projector 2.

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

We must not forget the often neglected management aspects of projects and products. Also these subjects must be further researched. At UNU/IIST we do it in the light of the Design Calculi Programming Methods.

## The Methodology — continued: Mgt.

Project Management

9. Resource Estimation
10. Resource Budgeting
11. Resource Acquisition
12. Resource Allocation & Scheduling
13. Resource Deployment
14. Resource Monitoring & Ctrl.
15. &c.

Product Management

16. Market Analysis — See item 1
17. Product Determination — See item 2
18. Product Financing
19. Product Launch
20. Sales & Service
21. &c.

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

Foil 40 on projector 1, foil 38 on projector 2.

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

In all of this talk I have only illustrated some results of constructing domain analyses. Many facets of doing domain analysis still need by researched.

Perhaps Kyushu/ISIT might be interested in collaborating with UNU/IIST in this area?

# D/A: [Application] Domain Analysis

Informal & Formal Descriptions of all Domain Components (State & Input/Output, Structure/Composition)
and Component Functions, Operations & Behaviours

1. For each Domain "View"
   - (a) Synopsis — Informal
   - (b) Terminology — Informal
   - (c) Narrative — Informal
   - (d) Model — Formal
2. Domain "Views"
   - (a) Intrinsics
   - (b) Support Technology
   - (c) Policies: Rules & Regulations
   - (d) Staff & Client Behaviours
   - (e) Market, Economics — *&c.*
3. Analogy
   - (a) Maxwell's Equations — vis-a-vis
   - (b) Radio/TV Communication
4. Laws of Infrastructures

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

Foil 40 on projector 1, foil 41 on projector 2.

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

Similarly we pursue requirements capture as a process depending on but different from domain analysis. Again we practice the shown approach and further research the area. And again we invite collaboration.

# R/C: Requirements Capture

1. For each Requirements "View"
   (a) Synopsis
   (b) Terminology
   (c) Narrative
   (d) Formal Model
2. Requirements "Views"
   (a) System Identification
      i. External vs. Internal [System] State
      ii. Sensors & Actuators
      iii. System State Input & Update
   (b) Fcts, Opns, Behvs. Supported
      i. Invariants — see item 2(a)iii
      ii. Observers, Generators
      iii. Policies
   (c) Meta-Properties
      i. Distributivity
      ii. Reactiveness & Real-time issues
      iii. Safety Criticality & Dependability
      iv. CHI: User Interface
   (d) Computing & Comm. Platform

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

Foil 42 on projector 1, foil 41 on projector 2.

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

Finally, in our steps of development — of which we today only have shown the first, the domain analysis step — we show the concerns of the software architecture step. We understand software to mean all those behavioural facets of the software that users can observe — the "macro" — and not the Program Organisation — the "micro architecture" — such as currently advocated by especially researchers in the US.

# S/A: Software Architecture

1. The "Macro-Syntax" of Architectures:
   Software Assets
   (a) Command Engine
   (b) Display Engine
   (c) Compute Engine
   (d) Network Engine
   (e) Database Engine
   (f) Transaction Processor
2. The "Micro-Syntax" of Architectures:
   Software Devices
   (a) Rôles
   (b) Connectors/Connections
   (c) Configurators/Configurations
   (d) Allocators & Schedulers

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

| |
|---|
| Foil 43 on projector 1. |

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

| |
|---|
| UNU/IIST works with Fellows from and institutions in developing countries on all these and other projects. We do so in small, manageable, 8–10 months phases and with professional delivery of results. |

# R&D Project Organisation

## Collaboration with Public/Private Sector

1. Phases:
   (a) Feasibility
   (b) Exploratory
   (c) Demonstrator
   (d) Technology Transfer
2. Deliverables
   (a) Target Devt. Staff Training
   (b) Target Management Training
   (c) Target Sector Training
   (d) Full Documentation
   (e) Prototype Software
   (f) Software House/Product Business Plan
   (g) Univ. M.Sc. Candidate Training + Orientation
   (h) Univ. Ph.D. Candidate Training + Orientation
   (i) Science & Technology Reports
3. Joint Management Committee

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

Foil 44 on projector 1.

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

What is UNU/IIST: It is part of the UN University whose head quarter is in Tokyo and generously supported by the Government of Japan. It is but one of several research and post-graduate and post-doctoral centers spread around the world. It is the only one concerned with software technology. We give advanced courses all over the world — in developing countries. This year alone in: China, India, Vietnam, Argentina, Brazil, Turkey, Romania, Ukraine, Russia, Belarus, Philippines, Indonesia, Thailand, and Gabon. At these usually two week courses we identify 2–3 young and bright Fellows, discuss joint R&D with them and their colleagues, and invite them to Macau for 6–10 months! Here we then do either research into foundations of Design calculi or the advanced application of such Design Calculi — of which you have seen one being applied in todays talk. The R&D thus goes hand in hand. So far we have taught around 1,000 graduates and PhD-s in the last 4 years worldwide. We have had more than 40 fellows in Macau and presently have 14 such fellows. By September this year we will have more than twenty fellows full time! To tutor and do joint R&D with these fellows we have then a staff of 8 senior scientists + occasional visiting scholars. To round of the picture of UNU/IIST we occupy some 1,000 square meters on the 17th and 18th floor of a downtown Macau building and will move to a 2,500 square meter beautifully restored 100 year old villa next year. We have 30 million US Dollars in the bank and derive around 2 million US Dollars from that endowment fund for annual budget. In addition we receive contracts from developing countries, The World Bank, etc. To help us with all this we have a local General Services staff of 8 people.

# Why UNU/IIST's Interest?

1. Infrastructures: A UN Concern
   — and of Intl. Devt. & Reconstruction Banks
2. Programming in the Very Large:
   (a) From Packages to Systems
   Past: "patched" Transitions/Growth
   (b) No Prior Research Into Infrastructures
   New Programming Method: Domain Analysis (D/A)
   (c) Domain Analysis vs. Requirements Capture:
   (d) Requirements Capture vs. Software Development:
3. Research Issues: See next!
4. UNU/IIST and Developing Countries: Self-reliance
   (a) Industry Production
   (b) University Teaching
   (c) Research Institute
5. Own Software Industry in DCs:
   *Walking on two legs:*
   Domestic Consumption and International Export

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

| Foil 44 on projector 1, foil 45 on projector 2. |
|---|

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

In particular we like to engage ourselves and our Fellows in the Advanced Development projects because there is such a rich world of research topics to be studied and of problems to be resolved. The aim is, of course, to make software development far more efficient, reusable, of high quality and lower cost; more manageable, and certainly much more exciting and fun!

Among the research issues are the relations to other disciplines: Operations Research, Control Theory, Industrial Engineering, etc.

# Research Issues

1. Domain Analysis — NEW
2. Requirements Capture — NOVEL
3. Software Architectures — DIFFERENT
4. A Domain Meta-Theory:
   (a) Intrinsics $\otimes$ Supp.Techn. $\Rightarrow$ Plant
   (b) Policies $\otimes$ Plant $\Rightarrow$ Deployment
   (c) Staff $\otimes$ Deployment $\Rightarrow$ Operations
   (d) Client $\otimes$ Operations $\Rightarrow$ Business
   (e) Economy $\otimes$ Market $\otimes$ Business $\Rightarrow$ Profit
   (f) C&C $\otimes$ Profit $\Rightarrow$ Growth
5. Infrastructure Theories
6. Relations between Computing Science and:
   (a) Operations Research
   (b) Control Theories
   (c) Management Theories
   (d) Business Theories
7. Laws!

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

Foil 46 on projector 1.

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

It is time to conclude: what have we shown and what are we achieving?

We have introduced the concept of Semantics of Man-made languages, we have introduced the related concept of Infrastructures; we have indicated the importance of securing adherence to Laws, and we have put all this in the context of the specific Development Method practice and studied at UNU/IIST.

# Review & Conclusions

1. Semantics of Man-made Systems
2. Introduced the Infrastructure Concept
3. Laws
4. Exemplified Infrastructure R&D Projects
5. Reviewed the UNU/IIST Devt. Method
6. Laws

7. **Please Join UNU/IIST in this Research!**

# Notes for Interpreters, Projectionist and Speaker

**Notes to Interpreters :**

Foil 46 on projector 1, foil 47 on projector 2.

**Words Spoken by Speaker :** Instead of Text on Displayed Foils

Thank you most kindly for:

1. Honouring me through inviting me to come here and present this talk.
2. UNU/IIST wishes IAI a bright and successful future
3. UNU/IIST wishes bright and successful future CARI conferences
4. Thanks in particular to IAI and INRIA
5. Thank you for hopefully answering many questions now
6. I am ready for questions and will do my best to answer your questions: