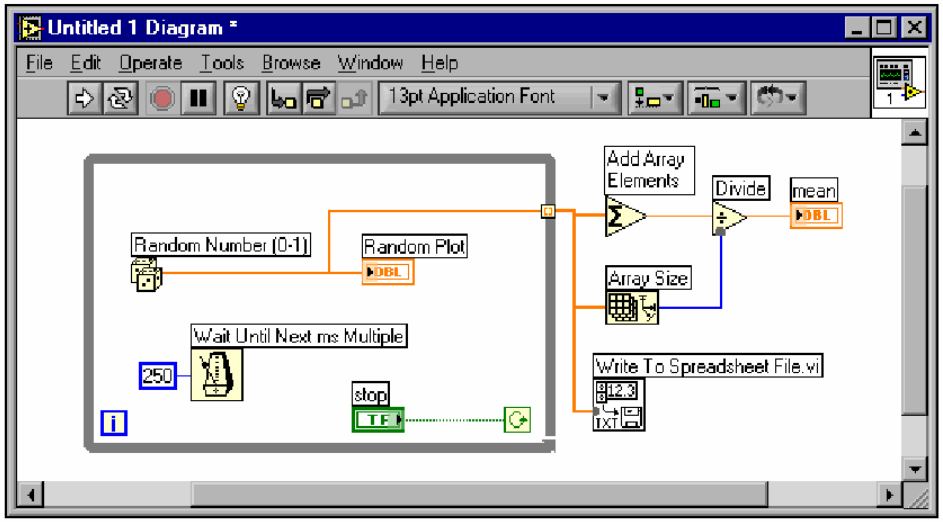
Domain Specific VLs

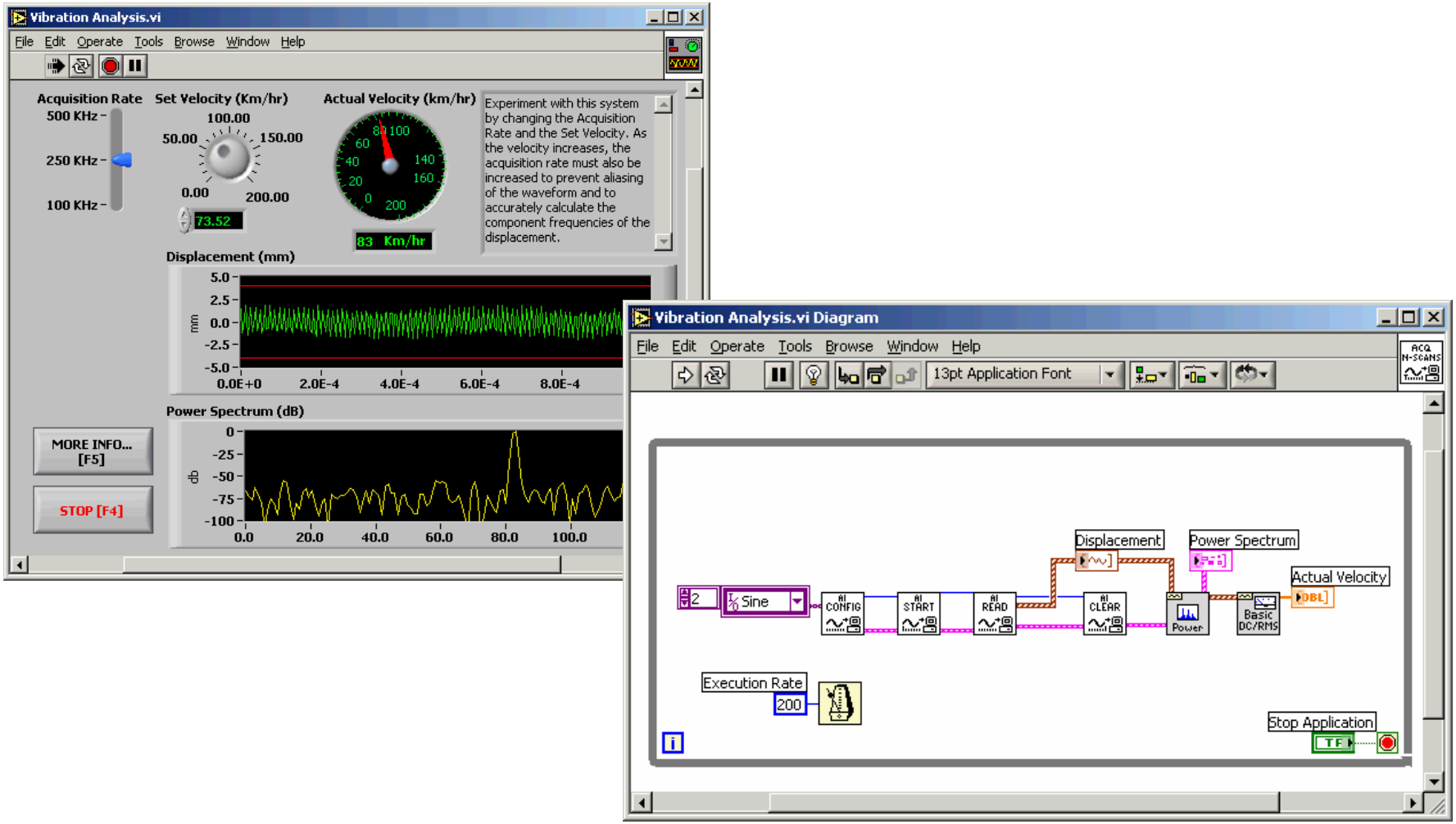
- A domain specific visual language is one where the notation is customised for a particular problem domain
- Have trade off between generality of language (ie range of problems able to be solved) and terseness of notation and closeness of mapping
- Look at:
  - A couple of widely used DSVLs
  - One that is likely to be widely used
  - Four locally developed DSVLs

LabView

- LabView uses a visual dataflow metaphor like Prograph, but is a domain specific language rather than a GP one
  - Domain is lab instrumentation: access and analysis of sensor data attached to computer
  - Processing elements include math data transformations (eg FFTs, integrators, differentiators)
- Very successful commercial Domain Specific VL
  http://www.ni.com/labview/



Labview example

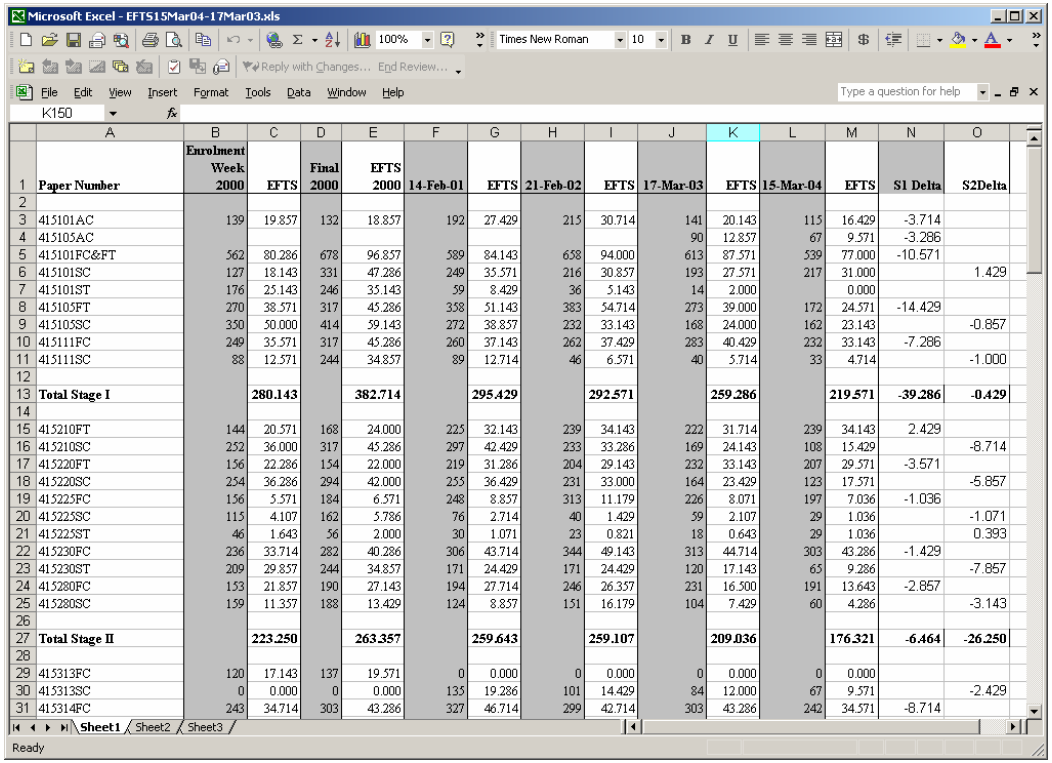


Labview Success

- Metaphor used – dataflow wiring plus computation blocks – has high closeness of mapping
  - End users are electronic engineers – very familiar with circuit wiring
- Modularity via blocks – again very similar to electrical circuit concepts hence low abstraction gradient for end users and hidden dependencies are of a sort that end users are familiar with
- Problems of high viscosity due to layout reorganisation not an issue with user audience – familiar with these problems from circuit design tools
- Language relatively terse at one level (general concepts) but quite diffuse at another (many predefined operations with their own iconic representation)
- Attention to front end – ability to create realistic looking virtual instrument front panel

Spreadsheets

- Very successful DSVL – so successful spreadsheets have become a more general tool
- Original target – financial and other numeric calculations
- Metaphor – financial tables + calculator



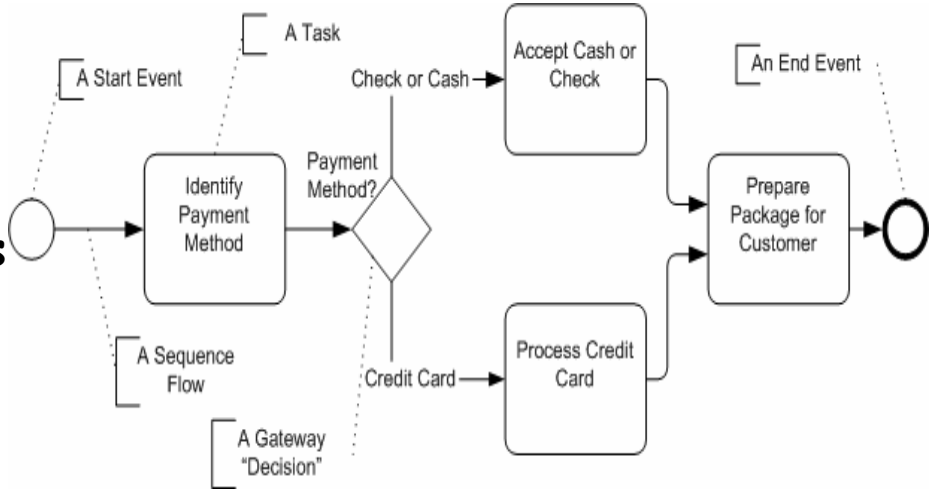
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Paper Number	Enrolment Week 2000	EFTS	Final 2000	EFTS 2000	14-Feb-01	EFTS	21-Feb-02	EFTS	17-Mar-03	EFTS	15-Mar-04	EFTS	S1 Delta	S2Delta
2															
3	415101AC	139	19.857	132	18.857	192	27.429	215	30.714	141	20.143	115	16.429	-3.714	
4	415105AC									90	12.857	67	9.571	-3.286	
5	415101FC&FT	562	80.286	678	96.857	589	84.143	658	94.000	613	87.571	539	77.000	-10.571	
6	415101SC	127	18.143	331	47.286	249	35.571	216	30.857	193	27.571	217	31.000		1.429
7	415101ST	176	25.143	246	35.143	59	8.429	36	5.143	14	2.000		0.000		
8	415105FT	270	38.571	317	45.286	358	51.143	383	54.714	273	39.000	172	24.571	-14.429	
9	415105SC	350	50.000	414	59.143	272	38.857	232	33.143	168	24.000	162	23.143		-0.857
10	415111FC	249	35.571	317	45.286	260	37.143	262	37.429	283	40.429	232	33.143	-7.286	
11	415111SC	88	12.571	244	34.857	89	12.714	46	6.571	40	5.714	33	4.714		-1.000
12															
13	**Total Stage I**		**280.143**		**382.714**		**295.429**		**292.571**		**259.286**		**219.571**	**-39.286**	**-0.429**
14															
15	415210FT	144	20.571	168	24.000	225	32.143	239	34.143	222	31.714	239	34.143	2.429	
16	415210SC	252	36.000	317	45.286	297	42.429	233	33.286	169	24.143	108	15.429		-8.714
17	415220FT	156	22.286	154	22.000	219	31.286	204	29.143	232	33.143	207	29.571	-3.571	
18	415220SC	254	36.286	294	42.000	255	36.429	231	33.000	164	23.429	123	17.571		-5.857
19	415225FC	156	5.571	184	6.571	248	8.857	313	11.179	226	8.071	197	7.036	-1.036	
20	415225SC	115	4.107	162	5.786	76	2.714	40	1.429	59	2.107	29	1.036		-1.071
21	415225ST	46	1.643	56	2.000	30	1.071	23	0.821	18	0.643	29	1.036		0.393
22	415230FC	236	33.714	282	40.286	306	43.714	344	49.143	313	44.714	303	43.286	-1.429	
23	415230ST	209	29.857	244	34.857	171	24.429	171	24.429	120	17.143	65	9.286		-7.857
24	415280FC	153	21.857	190	27.143	194	27.714	246	26.357	231	16.500	191	13.643	-2.857	
25	415280SC	159	11.357	188	13.429	124	8.857	151	16.179	104	7.429	60	4.286		-3.143
26															
27	**Total Stage II**		**223.250**		**263.357**		**259.643**		**259.107**		**209.036**		**176.321**	**-6.464**	**-26.250**
28															
29	415313FC	120	17.143	137	19.571	0	0.000	0	0.000	0	0.000	0	0.000		
30	415313SC	0	0.000	0	0.000	135	19.286	101	14.429	84	12.000	67	9.571		-2.429
31	415314FC	243	34.714	303	43.286	327	46.714	299	42.714	303	43.286	242	34.571	-8.714	

Spreadsheet success

- Strong and consistent metaphor providing high closeness of mapping to typical balance sheet etc problems
- At one level notation is quite terse (sheet and cell metaphor), at another it is quite verbose (extensive range of functions that stretch the bounds of the matephor)
- Progressive evaluation well supported: values calculated immediately a formula entered
- Hidden dependencies a real issue – a strong cause of errors, ie leading to error proneness

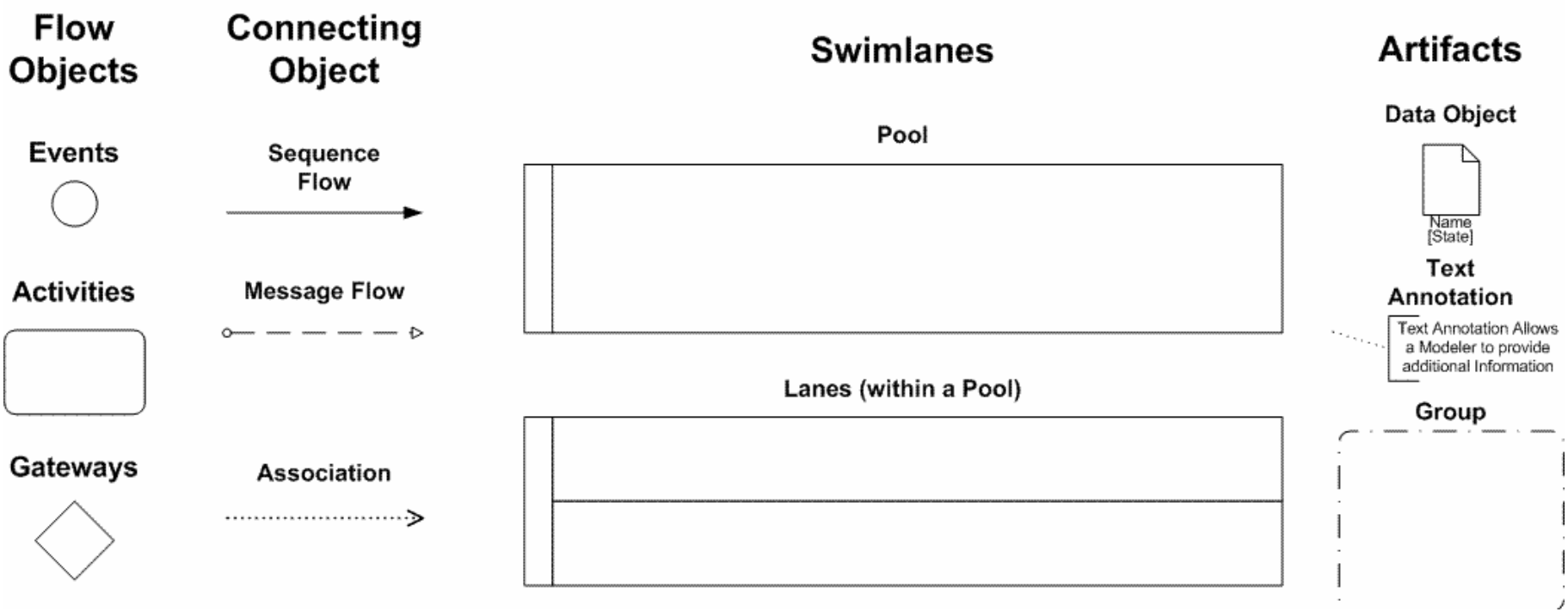
BPNM

- Business Process Modelling Notation
- Provides notation for specifying business processes
- Understandable by:
  - Business users
  - Business Analysts
  - Technical Developers
- Can generate BPEL4WS executable from BPMN specifications
- Developed by BPMI consortium http://www.bpmn.org/

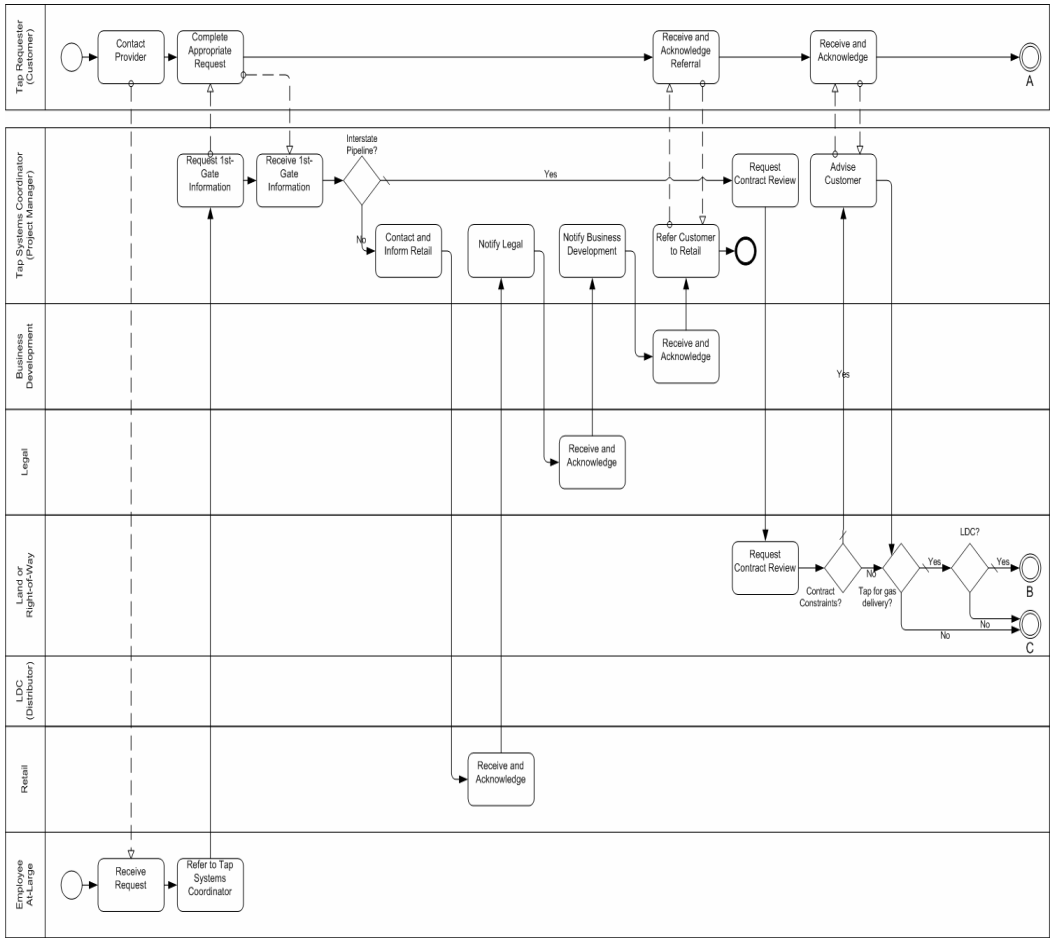


BPNM

- **Simple notation**



BPNM complex example



BPNM successes

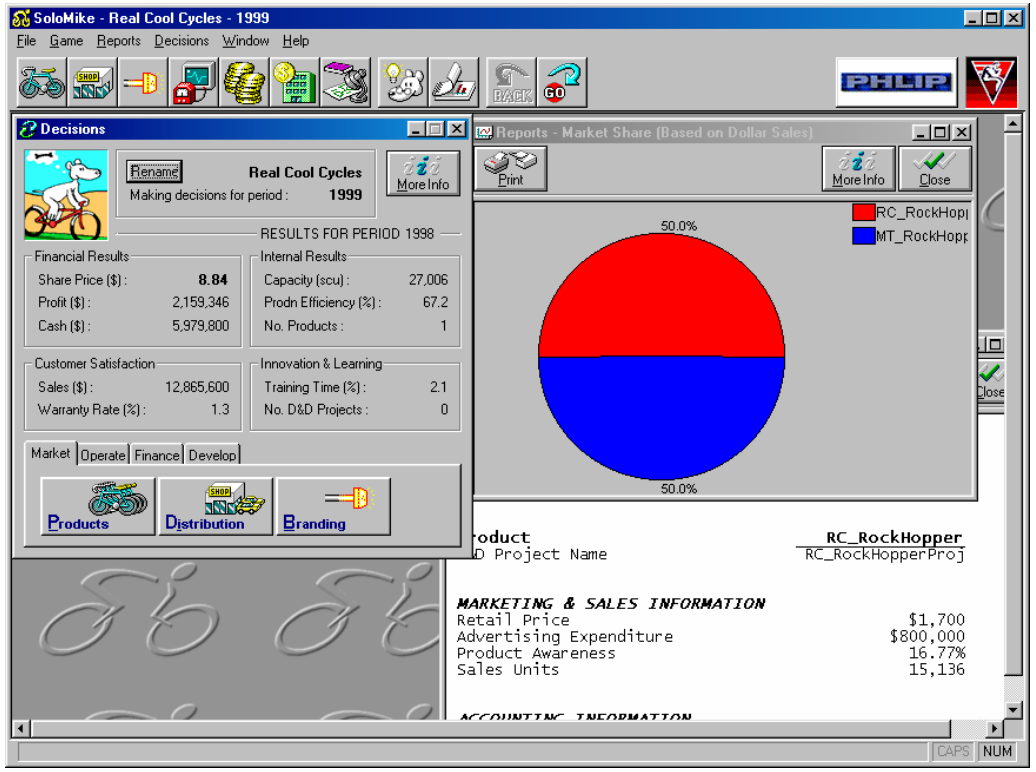
- Small number of concepts leads to a terse notation with small number abstractions. Good use of containers 9eg pools, lanes) to provide additional semantics without overloading user.
- Closeness of mapping high as core notation familiar to end users and business analysts (specific design choice).
- Good visibility and few hidden dependencies. Subprocesses can be "in lined" if desired.
- Low progressive evaluation as a design notation, but potential for runtime visualization reusing design diagrams

Local Work

- SmartSims Builder (business modeller)
  - Model business interactions for business simulation game
- Orion Mapper/Rhapsody (message mapping)
  - Mapping between different semi-structured data schema
- Form based mapper (business form mapping)
  - Mapping between business forms, eg invoices
- SDL (statistical survey specfn)
  - Specifying various stages in the design of a statistical survey

Mikes Bikes (SmartSims)

- Simulates bicycle business
- Factors in R&D, marketing, production costs, etc
- Play against robot or other teams
- Not generic: business model hard coded
- Need for matching business modelling tool to design new business situations
- MSc Thesis by See Wong

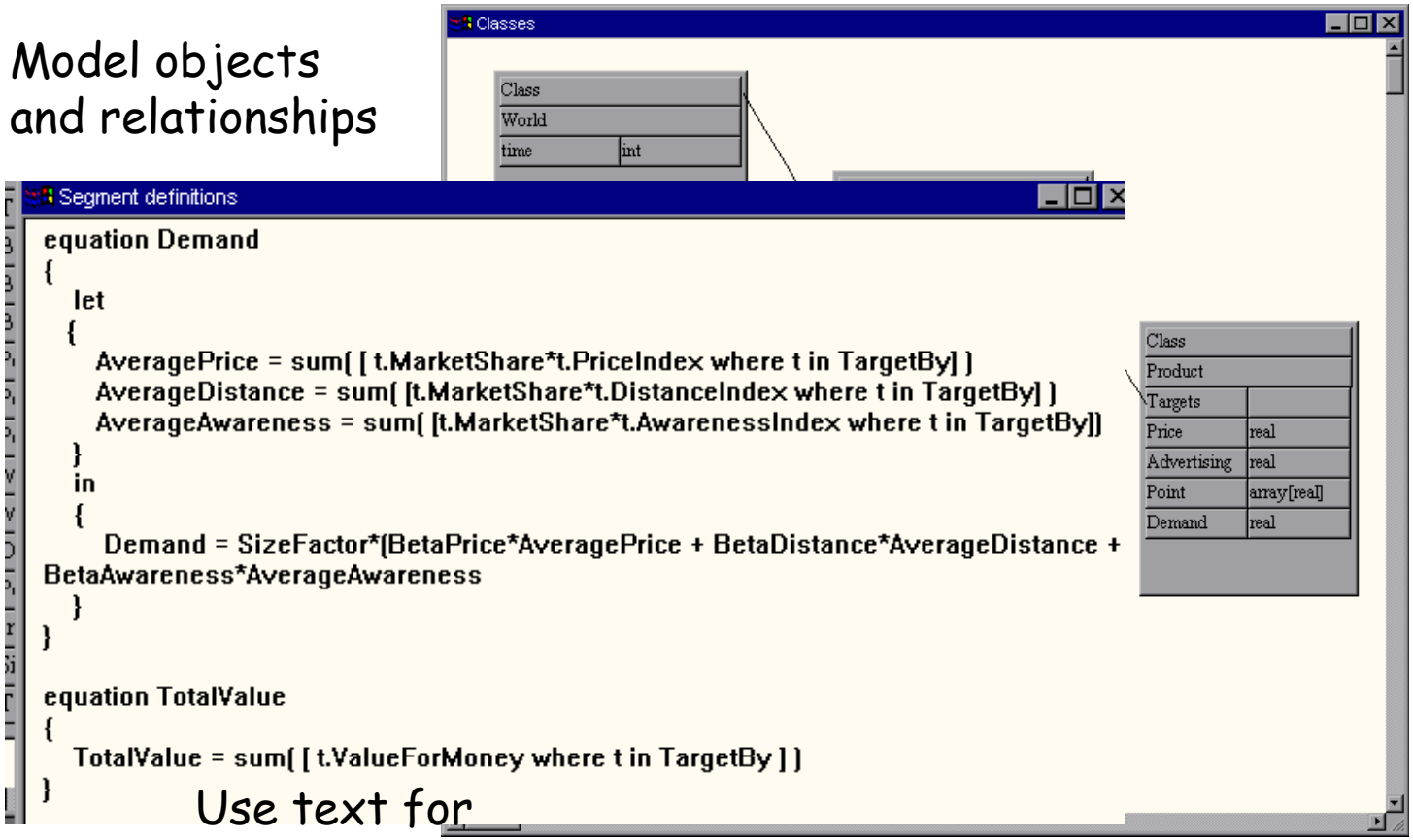


SmartSims

- Ideally would like business modellers to be able to construct new models
- Thus need a notation that has closeness of mapping for them
  - Familiar with entity relationship type modelling
  - Familiar with expression relationships using equations
- Thus combine the two using a UML like formalism
  - Something like a class diagram to represent business object types and their gross relationships (low abstraction gradient)
  - Reuse these with instantiated values to visualise enacted model (progressive evaluation/concreteness support)
  - Equational formulae to represent detailed relationships (combination of terse and verbose similar to spreadsheets)
  - GUI builder to construct UI of final simulation (cf Labview)

Builder environment

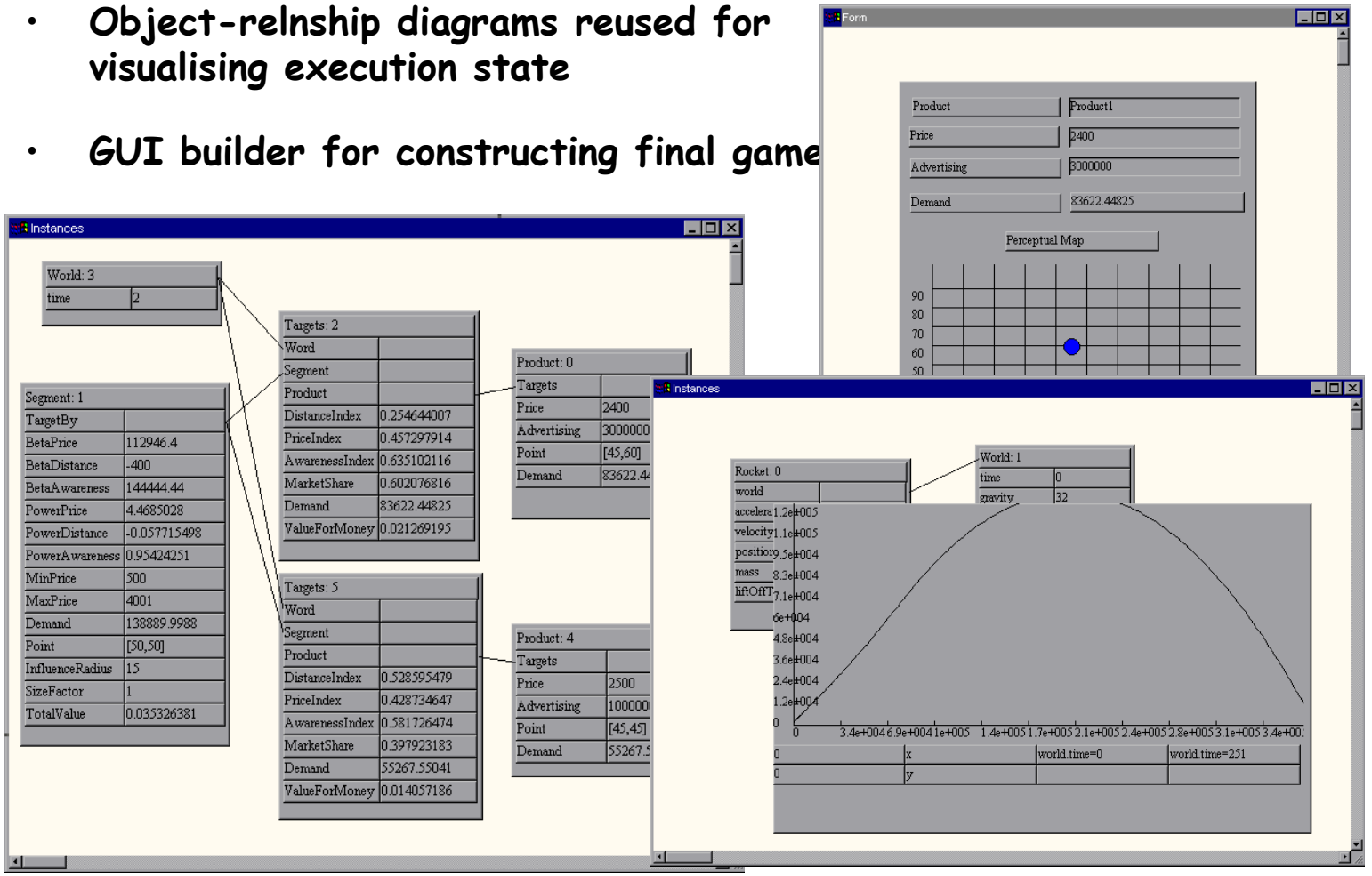
Model objects
and relationships



Use text for
expressing
constraint equations

Run time visualisation

- Object-relnship diagrams reused for visualising execution state
- GUI builder for constructing final game



Encore - Message Mapping (Orion Systems Ltd)

- **Basic problem: want to map semi-structured data (health messages) from one schema to another.**
  - Translating health information messages from one standard format to another involves laborious, time intensive, error-prone programming. Much of the code required is repetitive and hence lends itself to high level tool support with code generation
  - Many message standards plus XML-based variants
- **Typical current approach**
  - **C++ program, 10's of pages of code**
  - **Boring and tedious work - very error prone**

Encore - Message Mapping

Goals:

Design notations and implement a proof of concept tool to support complex hierarchical message translation (target is health sector, but wider application eg XML – XML translation is obvious)

User is experienced data modeller

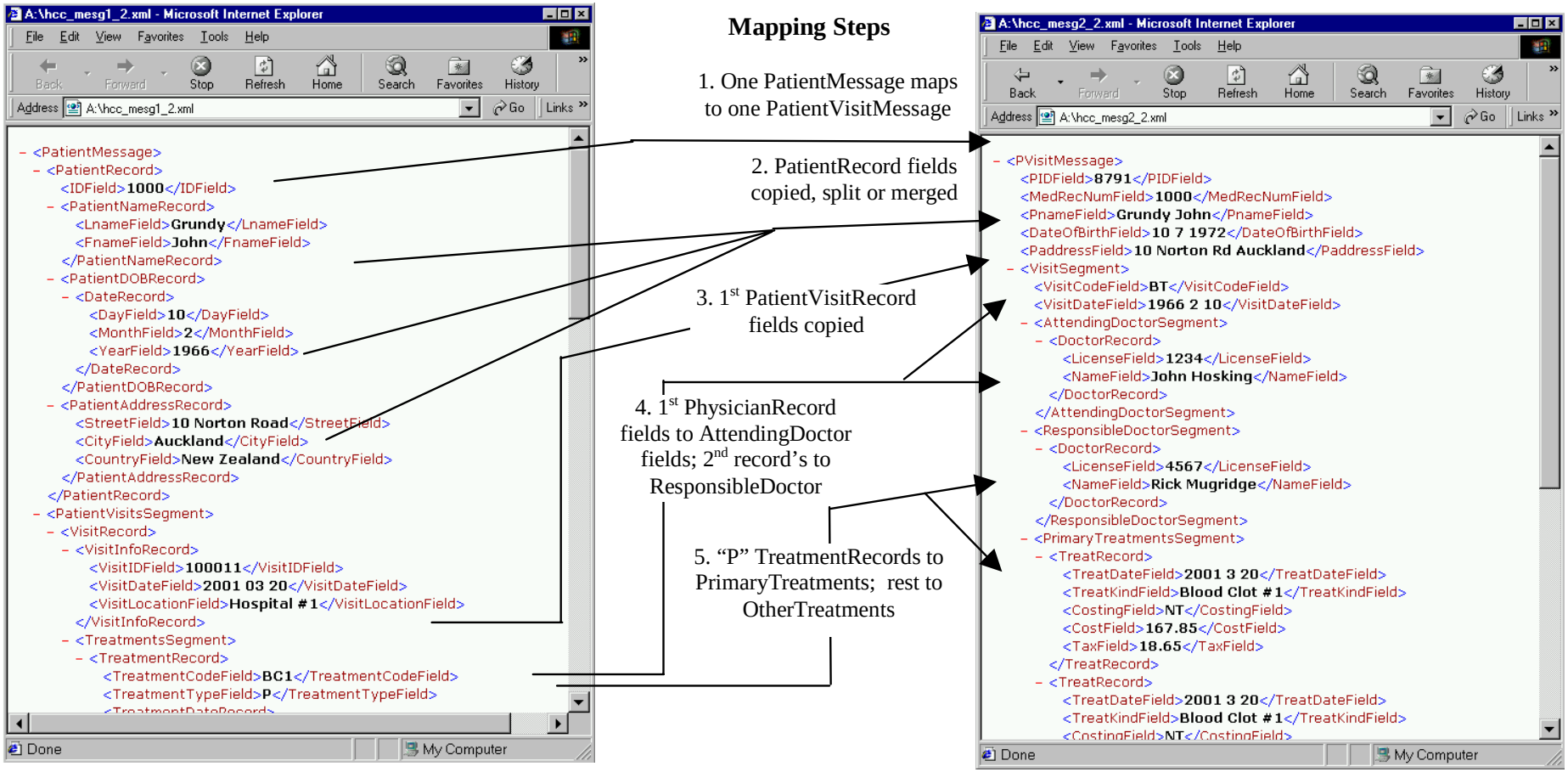
One component of a more complete approach to this problem (includes schema specn, message field formatting, DOM support, etc)

Solution

- Use a visual language to specify mappings between elements in hierarchical schemas (familiar notation for data modellers -> Good closeness of mapping)
- Supplement with a textual equational language for specifying individual element mappings (reuse terse + verbose of spreadsheets)
- Mappings can be uni- or bi-directional
- Compile to a threaded interpreter to execute mappings

Example of messages that need mapping

Fields, records need splitting, combining
Hierarchies added, flattened etc

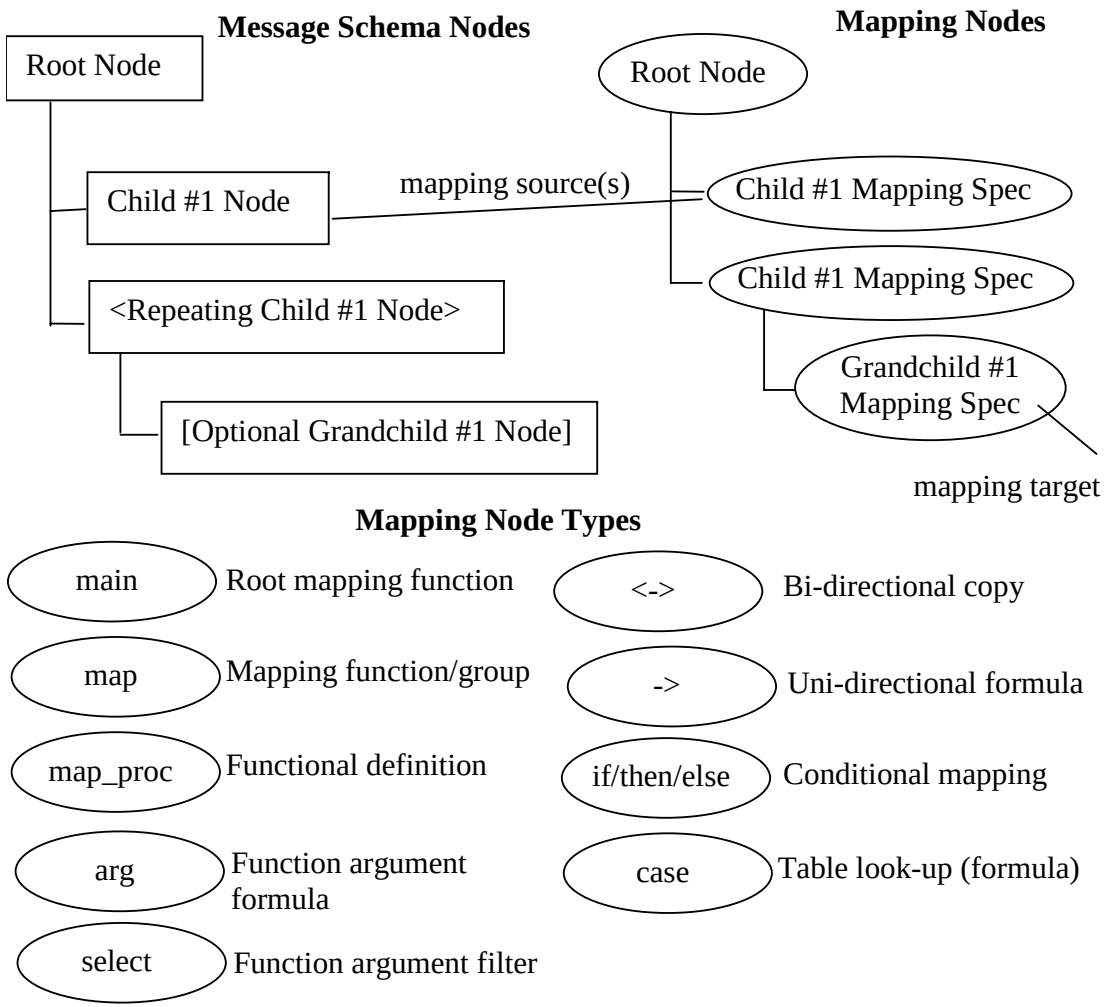


Basic visual notation for Mapper

Represent each schema vertically, with lines and indentation depicting hierarchy

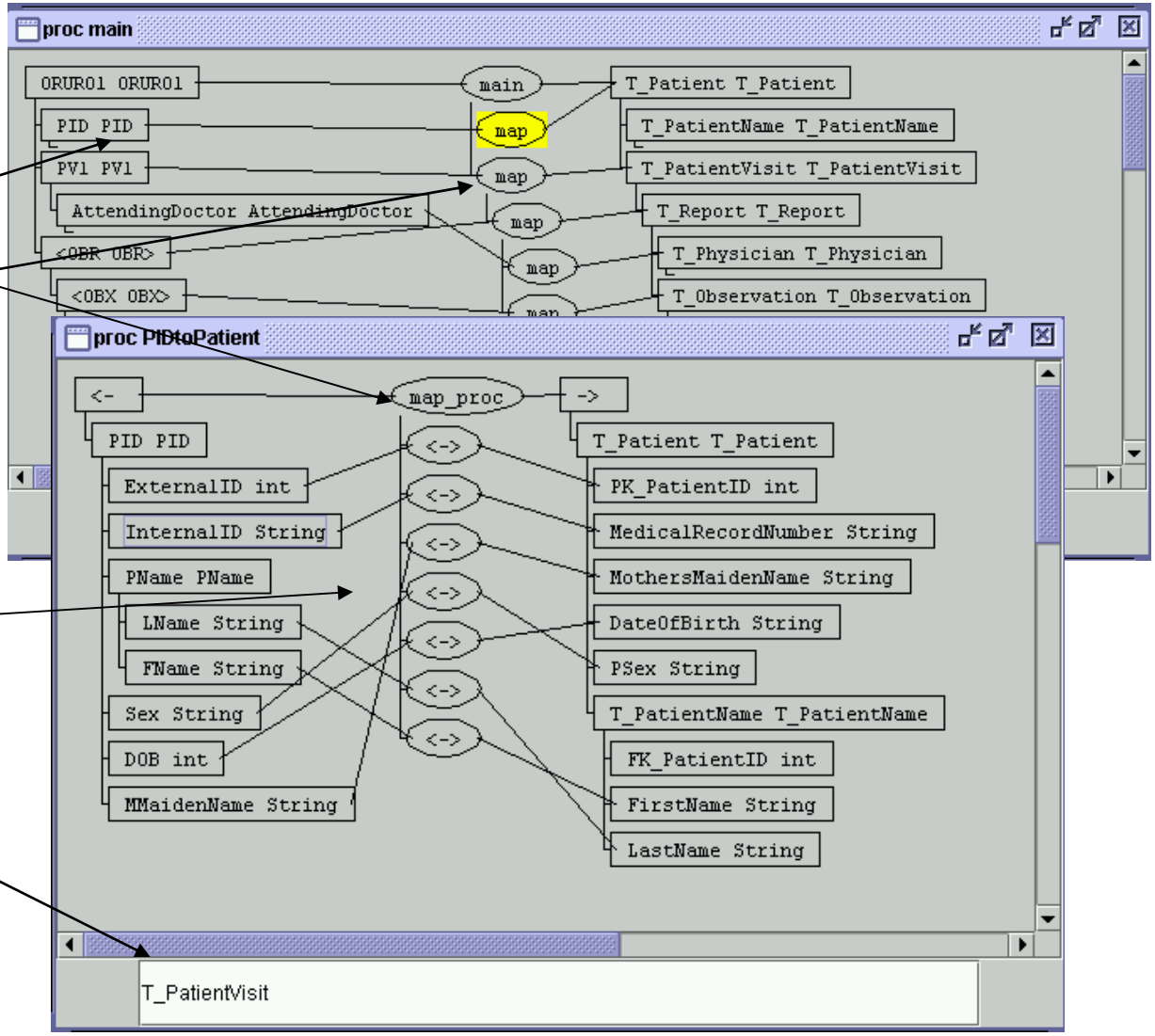
Mappings are functions between schema nodes

- simple copy
- translation formula
- conditional
- separate function with args
- table lookup
- etc



Proof of concept tool

- Mapping between hierarchies
- Proc abstraction map calls map procedure to map between patient name and PID records
- Equivalence mapping
- Textual formula box for more complex formula based mappings

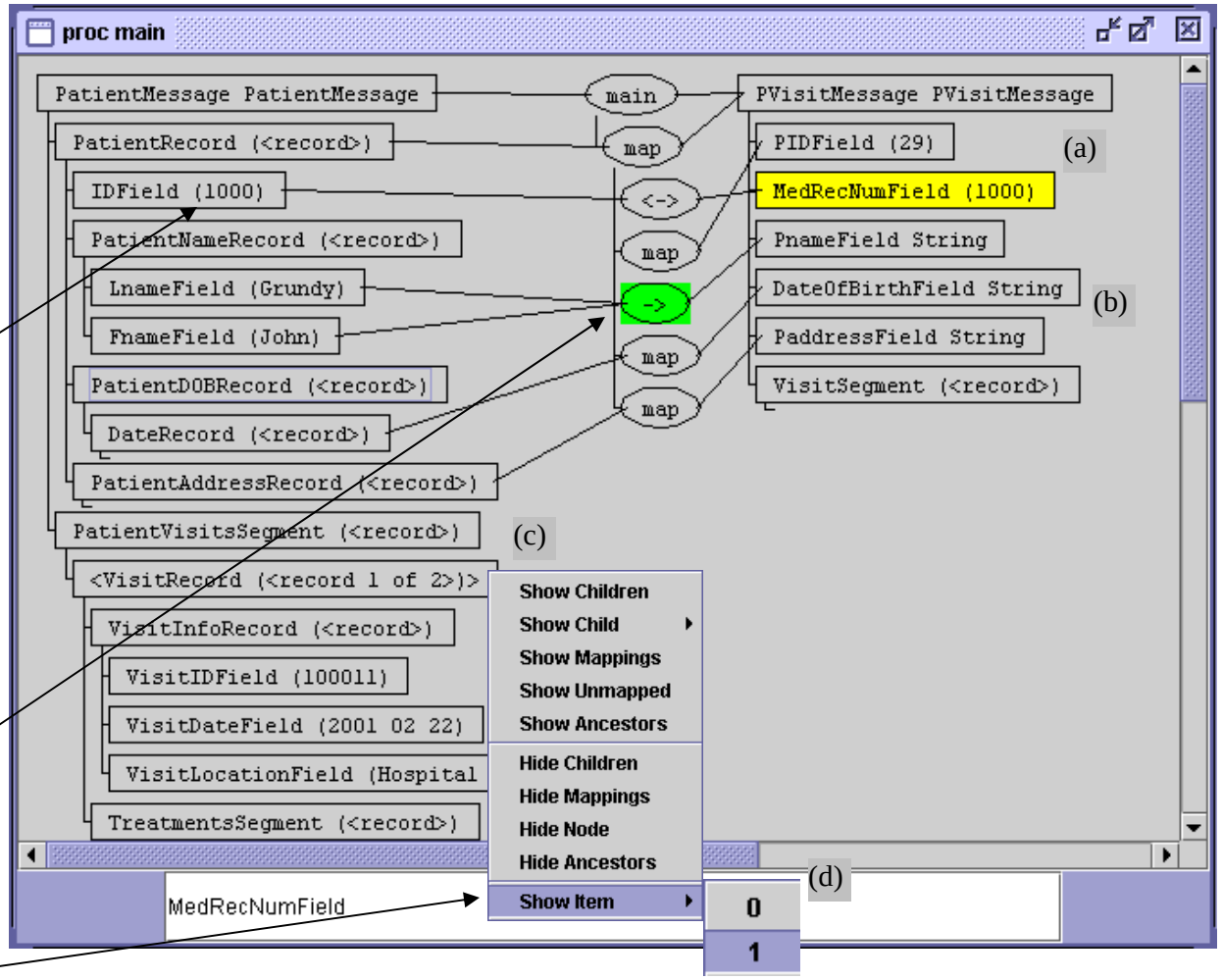


Mapping visualisation

Mapping diagrams reused to visualise execution (concreteness)

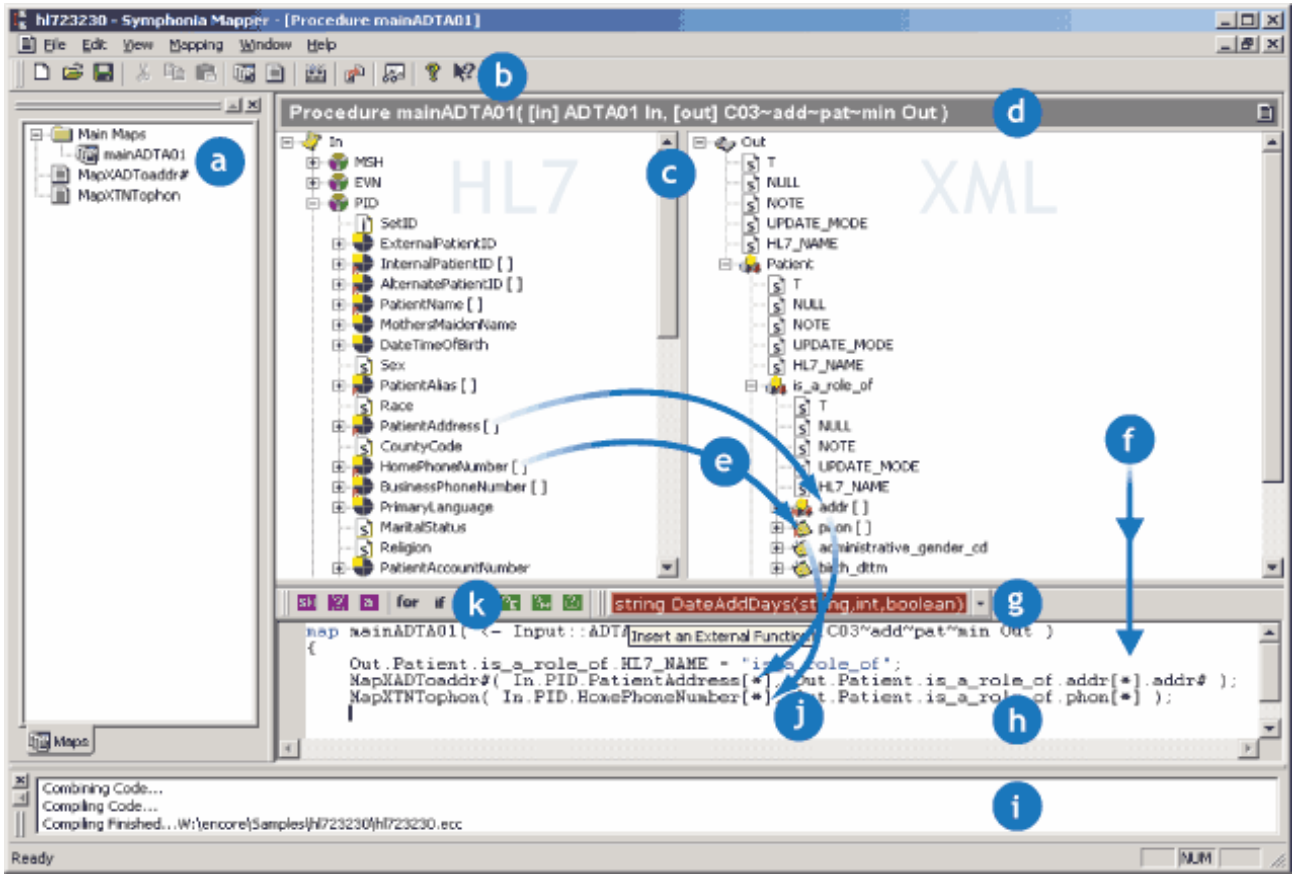
Hierarchies populated with values

Step through mapping execution and through collection values



Implemented using JViews framework

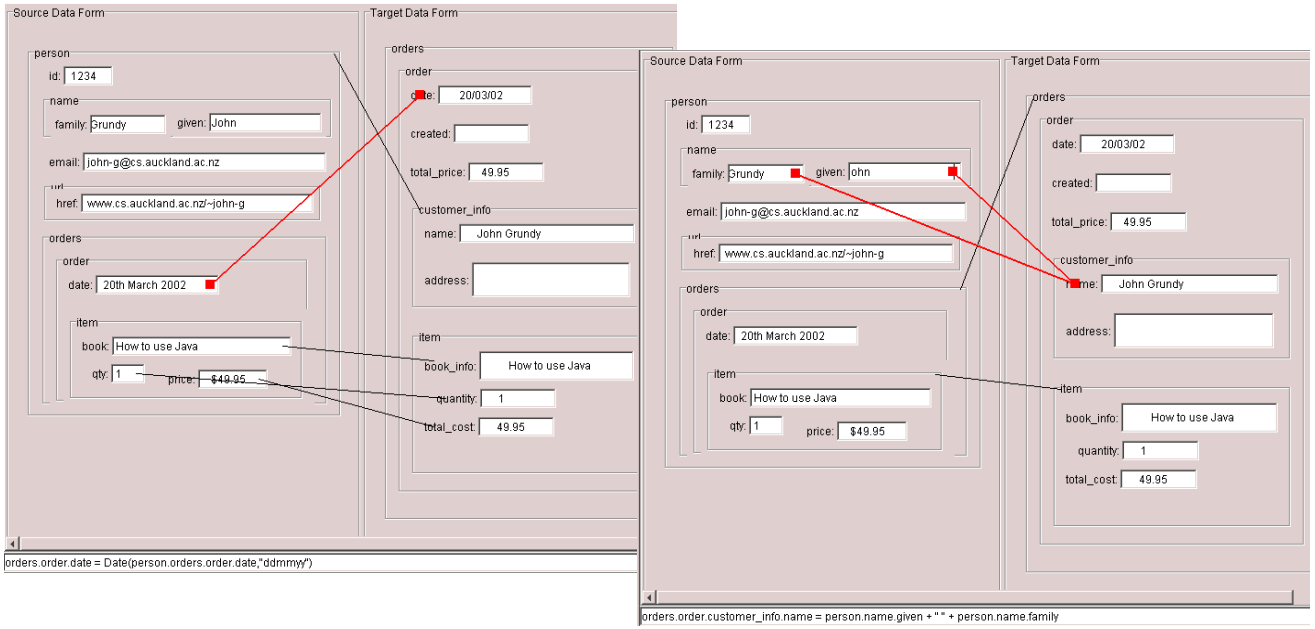
Commercial product: Rhapsody



http://www.orion.co.nz/

Form-based Mapper

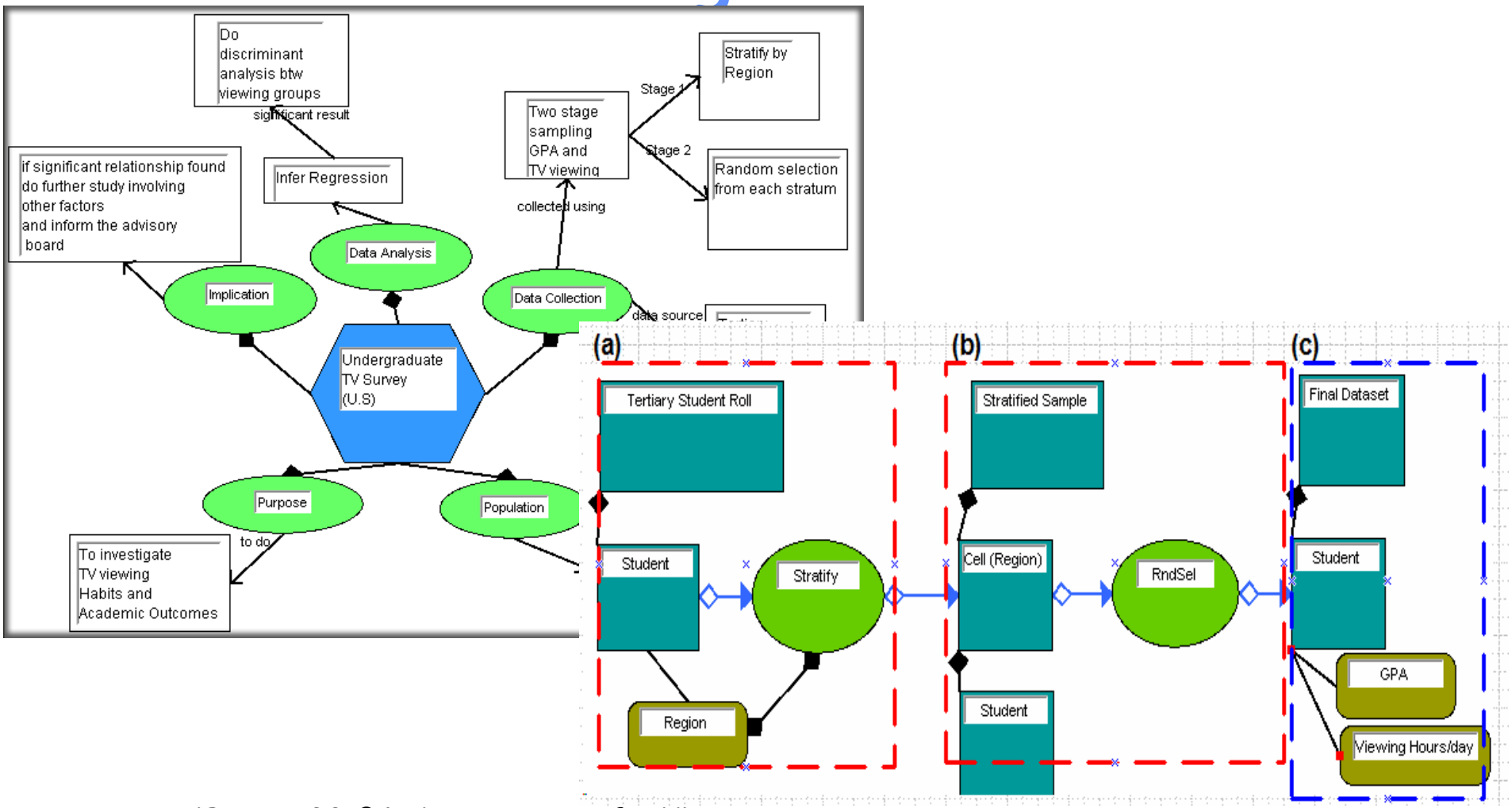
- An alternative approach to specifying mapping
  - Project by Yongqiang Li
- Aim to be used by a business modeller rather than a DBA
- Uses form metaphor rather than hierarchical tree



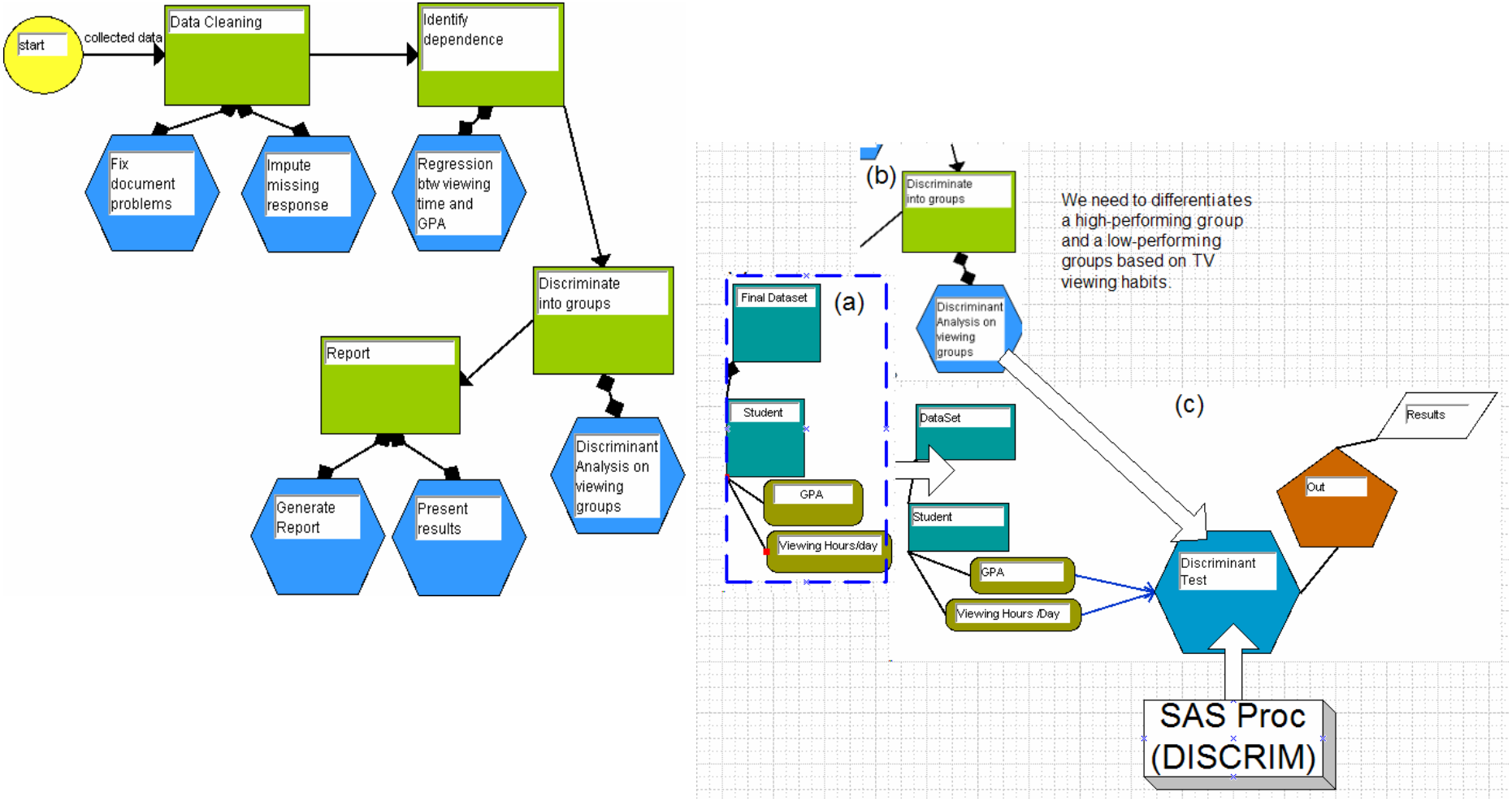
SDL

- Environment for specifying statistical surveys
- To be used by survey designers (not nec prof statisticians)
  - Tend to have minimal programming experience
- Solution
  - Provide multiple notations with consistency maintenance
    - Each notation has small number of concepts ie terse
  - Survey diagrams for brainstorming/overview
  - Survey data diagrams for survey source data structures and their manipulation
  - Survey analysis diagrams defining statistical processes/techniques and flow of control in data analysis phase
  - Survey techniques diagrams for specifying behaviour of individual statistical techniques
- Prototyped using Pounamu (originated as a 732 assignment!)

Survey and survey data diagrams



Survey analysis and technique diagrams



Summary

- Have looked at several Domain Specific VLs
- Emphasis in each case was to find a notation or set of notations that is in some senses natural for the end user (ie closeness of mapping rated highly as a cognitive dimension)
  - Function blocks + wiring for Labview
  - Table + calculator for spreadsheet
  - Object relnship diagrams + equational constraints for Builder
  - Tree or Form + drag and connect + formula for Encore/Mapper
  - Variety of metaphors for SDL
- Also emphasis on reusing diagrams at execution time to visualise behaviour at the same level of abstraction used to construct the program (moving towards liveness/progressive evaln and concreteness but recognising that compile cycle inevitable in many applications)
- Common to use terse high level abstractions and more verbose lower level detail (often textual) which gives some hidden dependencies and can lead to error proneness (cf spreasdsheets)

In-class Exercise

- Group of 4
- Assigned research paper
- EACH evaluate individually using Cognitive Dimentions
- Discuss & form consensus evaluation
- Prepare in-class presentation of results – 3-4 slides PPT MAXIMUM and 4min MAXIMUM
- Choose a presenter