process: something taking place
step-wise and in some organized manner
allowing for different realizations

the steps are the activities a process consists of
moreover, process consists of their organization
such as (partial) ordering, (partial) overlapping

so process is: activities + their organization

process model:
description of a process as
a set of its possible realizations
often abstract, visual, more or less formal

any formal description thereof is
a specification of process model,
and still a description of the original process

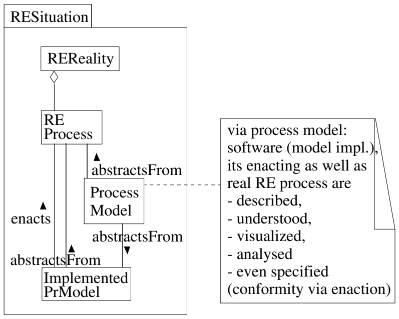
- the process model **never ever** is the process
but it describes it (abstractsFrom)

- its formulation specifies the process model,
not the process which it describes

- its implementation is a reformulation, perhaps
even refined (specialized) of the process model,
so it likewise describes the process
(abstractsFrom)

but: an implementation of a process model (PM)
can support the (real) process such that
the process is in conformity to the PM
in which case indeed
the PM formulation specifies the process too

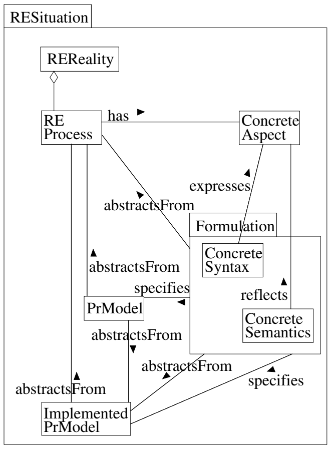
this is called "enacting" if support is complete,
ie each process step modelled and each organization of activities as modelled is being supported



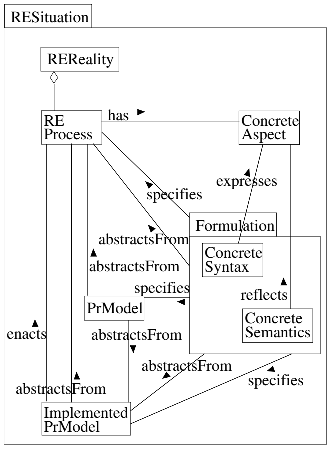
via process model:
software (model impl.),
its enacting as well as
real RE process are
- described,
- understood,
- visualized,
- analysed
- even specified
(conformity via enaction)

this is a special instance of the usual SE situation
where a model describes a real problem situation
as well as the solution to it offered by software

formal modelling language helps formal analysis



the formulation also **abstracts** both **from** real process and **from** implementation, as the process model it specifies, already does this
(formulation is phrase in a modelling language)

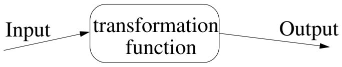


the formulation also **specifies** both real process and implementation, if via enaction (support)
conformity is assured

example process model of generic 1-step process

aka I/O model (Input-Output model) or
Black Box Model

in ICT: dataflow process-like



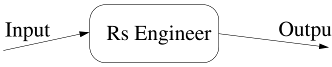
the idea is:

the activity of the transfer function
consists of

transforming the Input into the Output
by just saying
what is being transformed into **what**

instead of **how**, ie declarative, not operational

process model for RE process in particular



with I(nput):
- system info as-is, eg business (P)M
- all main process models
- stakeholders wishes
- organization standards, eg quality, RE practice
- rules, to check the above I and the eventual O
- domain knowledge, eg environment model

with O(utput):
- Rs doc
- particularly: Rs as officially approved
- software system specs: for designers
- software system models from different
perspectives: for designers

this RE process model is most generic

it still has to be made more specific by
specializing / refining / customizing it
(and possibly some form of instantiating)

among other things, this depends on client's

- technical maturity / ability
- disciplinary involvement / support-minded
- organizational culture
- application domain

2.1. Process Model(s) for RE

coarse-grained: see above

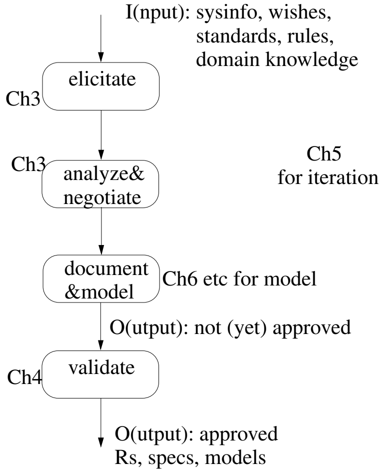
finer-grained: see below

book prefers: dataflow process-like
activities as transformational steps

also:
via relationships between
roles and activities: who does what

also: ER model giving the relevant structure

my favourite:
OOM (Object-Oriented Modelling)
as this comprises the above ingredients (aspects)



I(nput): sysinfo, wishes,
standards, rules,
domain knowledge

Ch3 elicitate

Ch3 analyze& negotiate

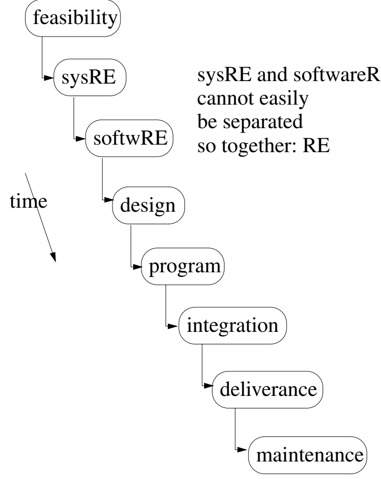
Ch5
for iteration

document &model Ch6 etc for model

O(utput): not (yet) approved

Ch4 validate

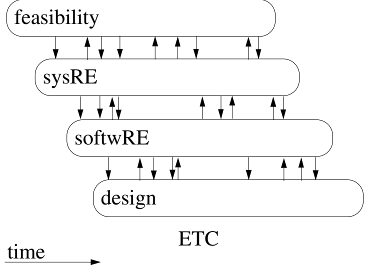
O(utput): approved
Rs, specs, models



feasibility
sysRE
softwRE
design
program
integration
deliverance
maintenance
time

sysRE and softwareRE
cannot easily
be separated
so together: RE

+ testing, gives iteration

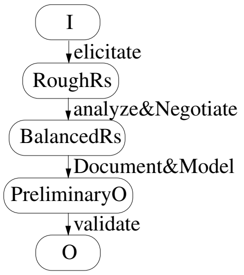


feasibility
sysRE
softwRE
design
ETC
time

there is substantial overlap,
giving natural room for feedback

instead of dataflow process-like PM from p 2.10:

STD-like PM (State Transition Diagram)



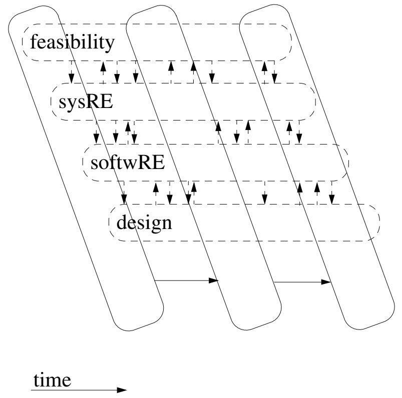
I
elicitate
RoughRs
analyze&Negotiate
BalancedRs
Document&Model
PreliminaryO
validate
O

I is: Input
O is: Approved Output

STD seems dual form of dataflow process

but: STD is strictly sequential, whereas
DFP can also be overlapping (parallel)

by grouping over all SE activities: prototyping

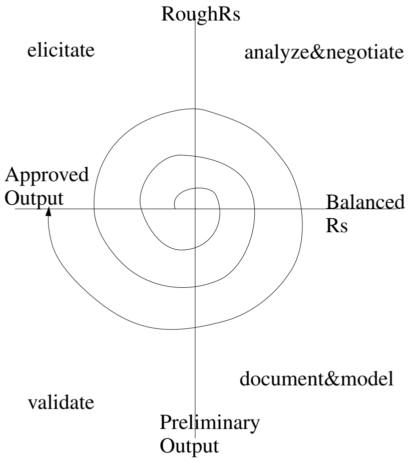


feasibility
sysRE
softwRE
design
time

visualizes incremental, evolutionary growth
of the prototype, thereby covering
all RE activities (phases)

also possible: covering the RE subactivities only

feedback as on p 2.12 and also on p 2.14
is expressed more explicitly via spiral model



RoughRs
elicitate
analyze&negotiate
Approved Output
Balanced Rs
document&model
validate
Preliminary Output

2.2 Actors / Stakeholders in RE process

Table 1:

Actor	Elic	AnNeg	DocMod	Val
REngineer	X	X	X	X
*DomainExpert	X			X
*(Business)User	X	X	X	X
*BusinessManager		X		X
ProjectManager			X	X
SoftwareEngineer			X	X
QualAssEngineer				X

*: via these stakeholders many influences of
non-technical character

- social
- psychological
- political
- organizational

2.3. Process Support

tools and techniques
- CASE
- SPE
- OOM

CASE: Computer Aided Software Engineering
- modelling tool (wrt the software-to-be)
- requirements DB

SPE: Software Process Environment
- as CASE plus
- SE process model (so RE process model too)
- more proactive than reactive

support in terms of

- modelling: drawing, storing, checking,
  possibly animating, rarely verifying

- document handling

- version management thereof

- (sub)activity management
  eg. workflow management
  or general enactment

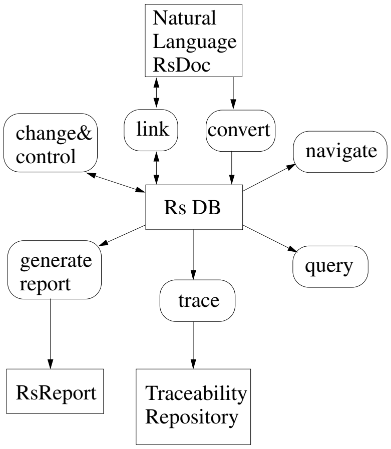
can be
- enforcing
- optional
- suggestive

but flexible tools for SE / RE are within reach

OO connects - integrates -
- structure
(statics, data: what things exist)
- behaviour
(dynamics: what steps take place)
- communication
(interaction: what influences what / how)
- possibly other (future) aspects too

note the integration: consistency is necessary

after Ch2
we'll look to UML 2.0 and OOM for RE



is too statical, defensive, inactive

we need moore active process support

on the basis of explicit PM
covering at least - apart from structure / data:
-behaviour, functionality,
- communication, coordination

such a PM enables
- not only observation / registration of
what happens
- but also control of what happens (next)

if "complete" aka enactment
ranging from
enforcement via suggestive to optional

2.4. Process Improvement

here it is part of
SPI: Software Process Improvement

what should improve:

- quality	(of product: here Rs)
	(of process: here RE process)
- time	(wrt process)
- resources	(wrt process, staff mainly)
- money	(costs of final product: reflects all)

so RE process as well as RE process model
have to change

preferably smoothly:
process evolution instead of process revolution

wrt planning of process improvement

- problem?
- goal?
- plan
- relation: goal vs planned improvement
- relation: goal vs established improvement

weak spots in every RE process (pit falls)

- stakeholder participation
no time, no interest, against it
- overlooking business needs
concentrating on mere technical matters
- no Rs management
no change of mind allowed
- no clear responsibilities
often mismatch between
insight and influence
- bad communication with stakeholders
different backgrounds, language

no standard set of process improvements exist

as this depends on
- problem
- organization:
problem domain, culture, maturity

but for SE we have (from DoD, SEI)
CMM: Capability Maturity Model

is related to ISO 9000

concrete relevance for business:
certification of actual maturity level

for RE process one can use similar model

SEI's CMM:

5 levels

1 Initial
-ad hoc, improvised
- not specified, not described

2 Repeatable
- implicit PM
- simple management of costs and of planning
- rough correspondence model and reality

3 Defined
- explicit model
- detailed correspondence model and reality

Up to level 3 is not exceptional
(although ...)
but level 4 and 5 are really difficult to reach
and to keep

4 Managed
- quality measurement of product and process
- process control on the basis thereof
towards improvement

5 Optimized
- explicit process model of
measuring - improving cycle
- detailed correspondence
model (of this cycle) and reality

for each level a fixed number of Key Practices
play the role of partial criteria fro the next level

improving then is:
introducing / installing more Key Practices

after having installed all KPs of a level,
one reaches the next level

wrt RE:

book proposes 3 levels:
1 Initial
2 Repeatable
3 Defined (+ Managed + Optimizing)

I prefer to refine this
precisely according to the original CMM

note:
role of explicit model of RE process

this is itself a business process
so we apparently do not hesitate to
model business processes explicitly and in detail

why no doing a similar thing for
general business processes
in the context of RE / SE?!:
see later: Integration-Orientation