Industrial Use of THE B METHOD

introducing B

- *a formal method …*
  - specification method based on a mathematical formalism to build models
- *… with refinement*
  - to structure a model step by step from abstract level to more detailed levels (and possibly to code level)
- *… and proof*
  - to prove that a model is consistent (in every possible case)
- *used for:*
  - system specification: B-System (or event-driven B)
  - software development: B-Software (or procedural B)

B and Atelier B: a formal method for

- *B for Systems*
  - Goal: help to understand, specify, design, verify a system development
    - not a method to create a system, but to check it
    - requires contacts with the system creators to deeply understand the system
  - a B-System model formalizes:
    - the system (hardware and software)
    - its environment (other systems, infrastructure, procedures handled by operators)
  - covers functional logical angle of the system, not digital calculus, not real-time requirements

CLEARSY
SYSTEM ENGINEERING

B and Atelier B: a formal method for

B for developing (safety-critical) Software

- Goal: to develop a code that complies with its specification and to be sure of it (to know exactly what is proved)
- a B-Software model formalizes the software itself, through a modules break down
- covers a subpart of the software with functional logical procedures, only for one task or thread, not low-level Operating System features, no direct input/output

CLEARSY
SYSTEM ENGINEERING

B-Software: Industrial References

- *KVB: Alstom*

  Automatic Train Protection for the French railway company (SNCF), installed on 6,000 trains since 1993

  60,000 lines of B; 10,000 proofs; 22,000 lines of Ada

- *SAET METEOR: Siemens Transportation Systems*

  Automatic Train Control: new driverless metro line 14 in Paris (RATP), 1998. 3 safety-critical software parts: onboard, section, line

  107,000 lines of B; 29,000 proofs; 87,000 lines of Ada

- *Roissy VAL: ClearSy (for STS)*

  Section Automatic Pilot: light driverless shuttle for Paris-Roissy airport (ADP), 2006

  28,000+155,000 lines of B; 43,000 proofs; 158,000 lines of Ada

CLEARSY
SYSTEM ENGINEERING

B-System: Industrial References

- *Peugeot Automobiles*
  - Model of the functioning of subsystems (lightings, airbags, engine, …) for Peugeot aftersales service
  - Goal: Understanding precisely the functioning of cars to build tools to diagnose breakdowns
- *RATP (Paris Transportation)*
  - Model of automatic platform doors to equip an existing metro line
  - Goal: Verifying consistency of System Specification

CLEARSY
SYSTEM ENGINEERING

B-System References

- *EADS*
  - Model of tasks scheduling of the software controlling stage separation of Ariane rocket
- *Study of a Communication Protocol*
  - Proof that the algorithm of a communication protocol complies with its requirements
- *INRS (French Institute for Workers Safety)*
  - Model of a mechanical press complying with safety requirements (protection of the hands of the press operator)
  - Building the software specification of the press controller

CLEARSY
SYSTEM ENGINEERING

basic concepts

- B is a method for specification (and possibly for programming)
  - B formalized system properties, static description, dynamic description
- B is based on a mathematical language
  - predicates, Booleans, sets, relations, functions
- B is structured
  - the notion of module
  - the notion of refinement
- B is a framework for development, validated by **proofs**
  - proof validation: systematic debugging

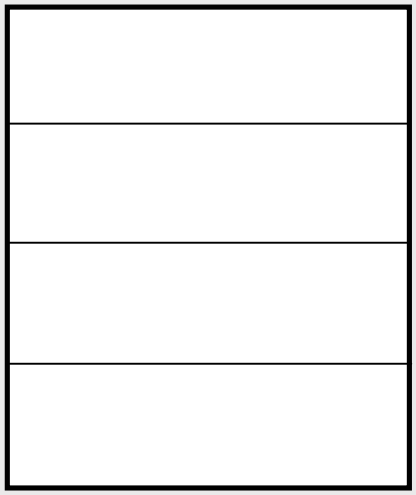
B structuring

- *the notion of modules*
  - to break down a large system or software into smaller parts
  - a module has a specification, where to formalize:
    - system properties
    - static description of requirements
    - dynamic description of requirements

B structuring

- *the notion of refinement*
  - a module specification is refined: it is reexpressed with more information:
    - adding some requirements
    - refining abstract notions with more concrete notions (design choice)
    - for B-software, getting to implementable code level
  - a refinement must be consistent with its specification (this should be proved)
  - a refinement may also be refined (refinement column)
  - for B-software, the final refinement is called the implementation

B structuring



module specification

module 1st refinement

module 2nd refinement

module 3rd refinement

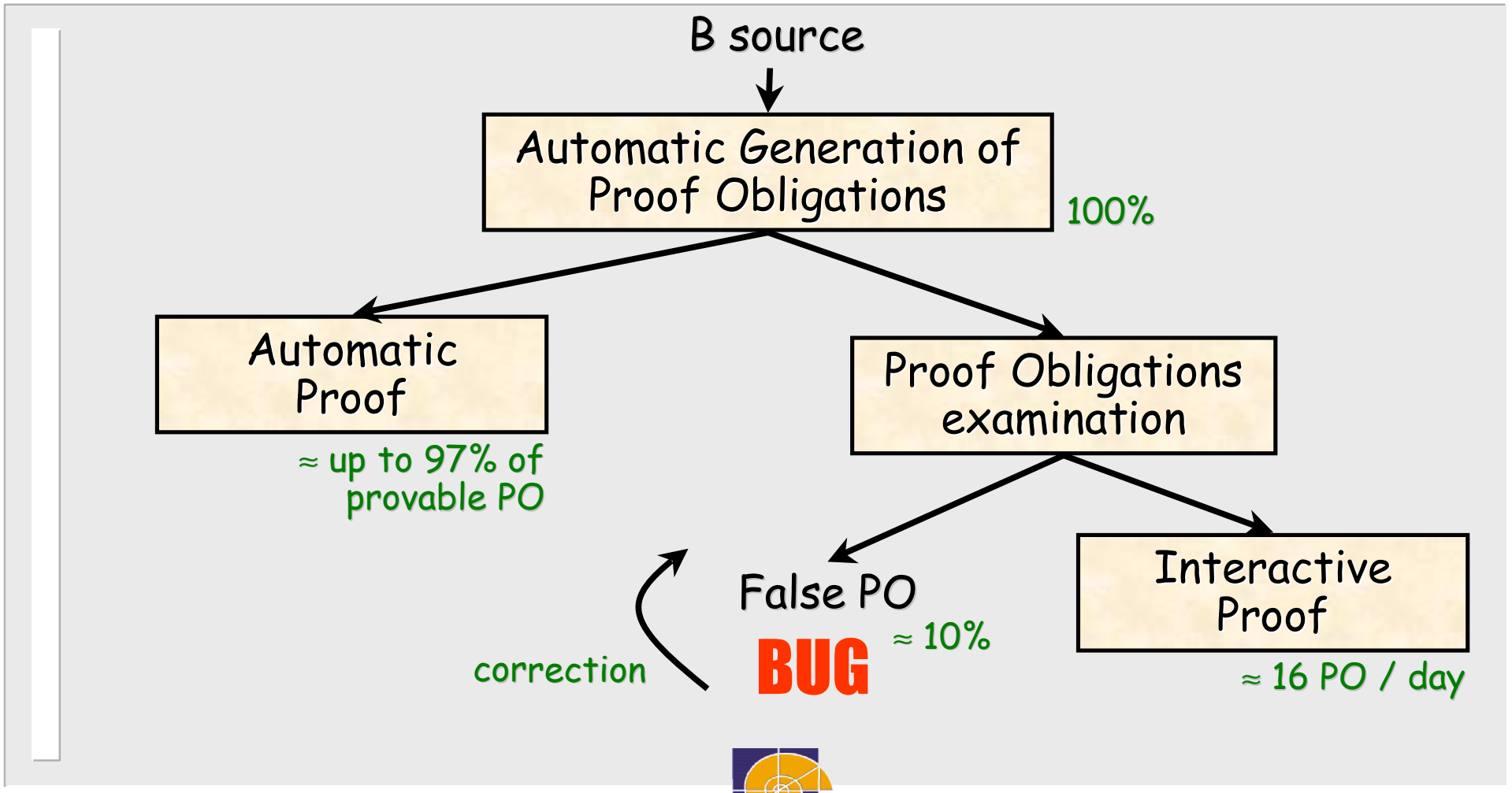
B module

B specification

Implementation or refinement

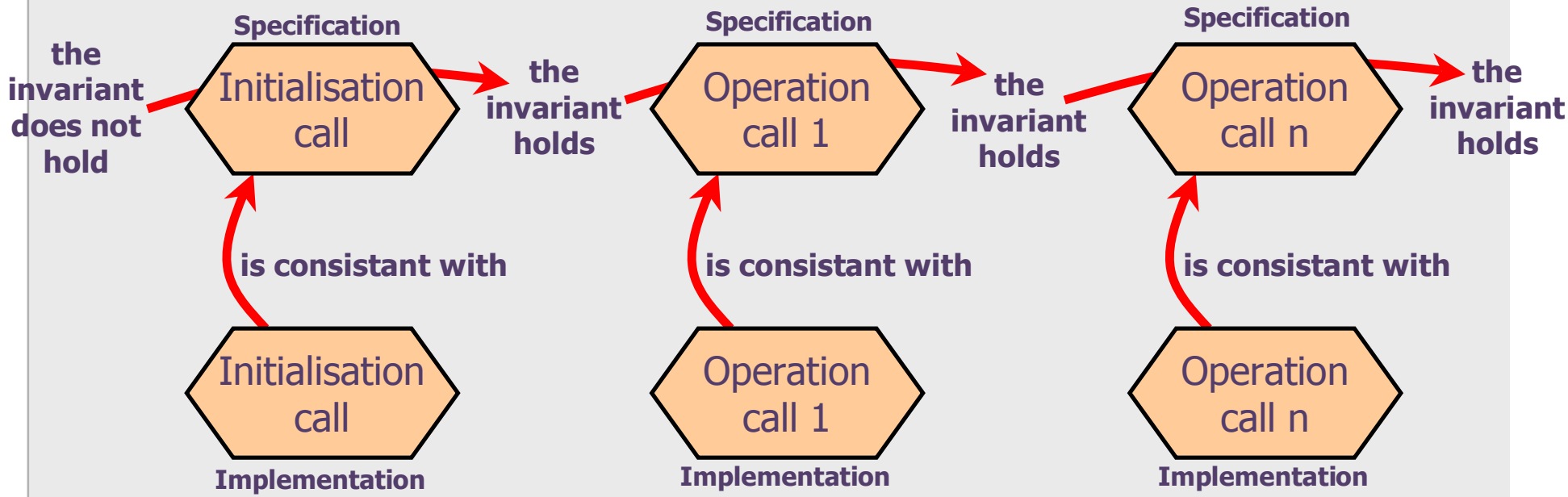
CLEARSY
SYSTEM ENGINEERING

validation by the proof



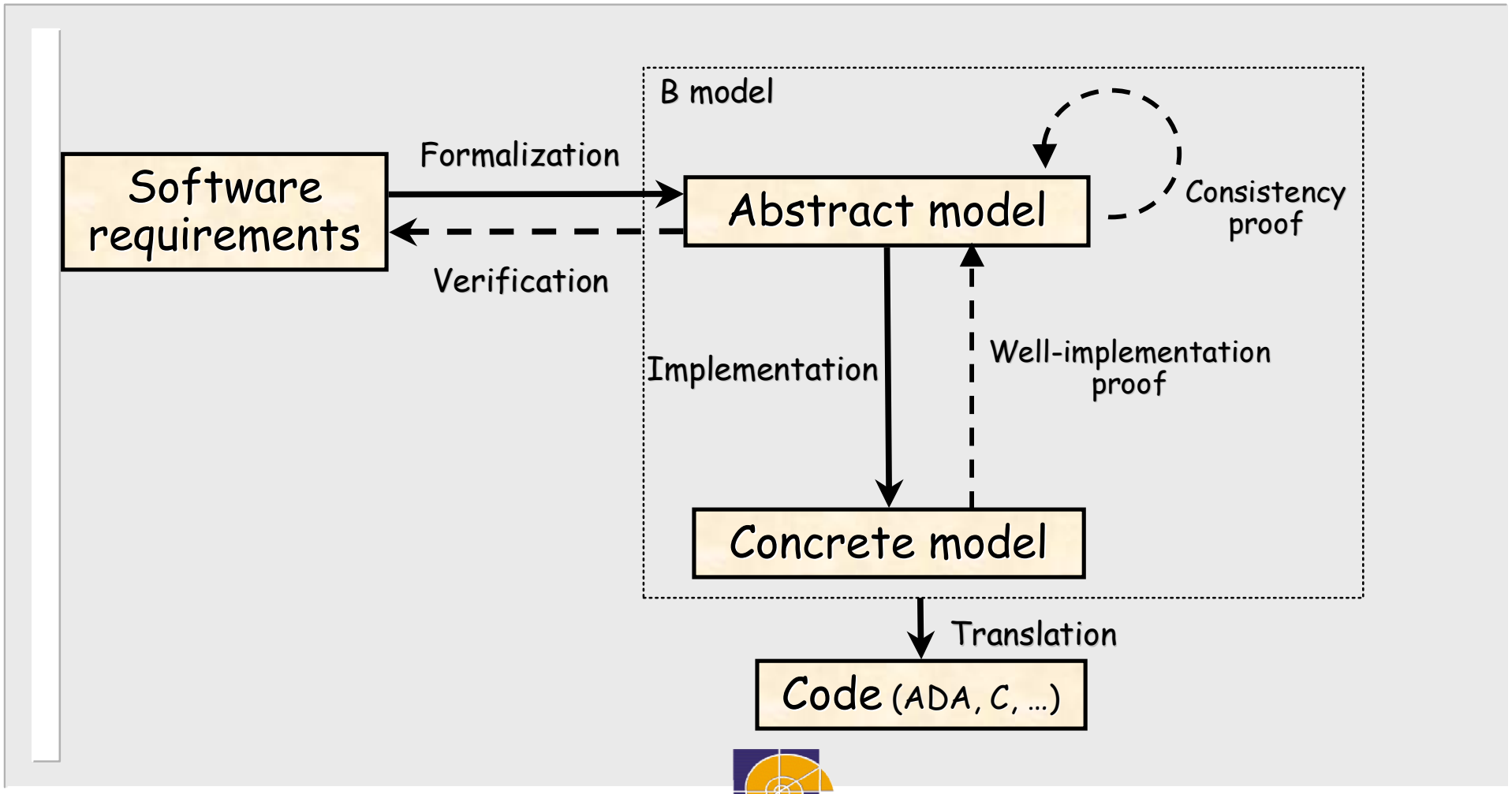
CLEARSY
SYSTEM ENGINEERING

Concepts of B: what is proved?



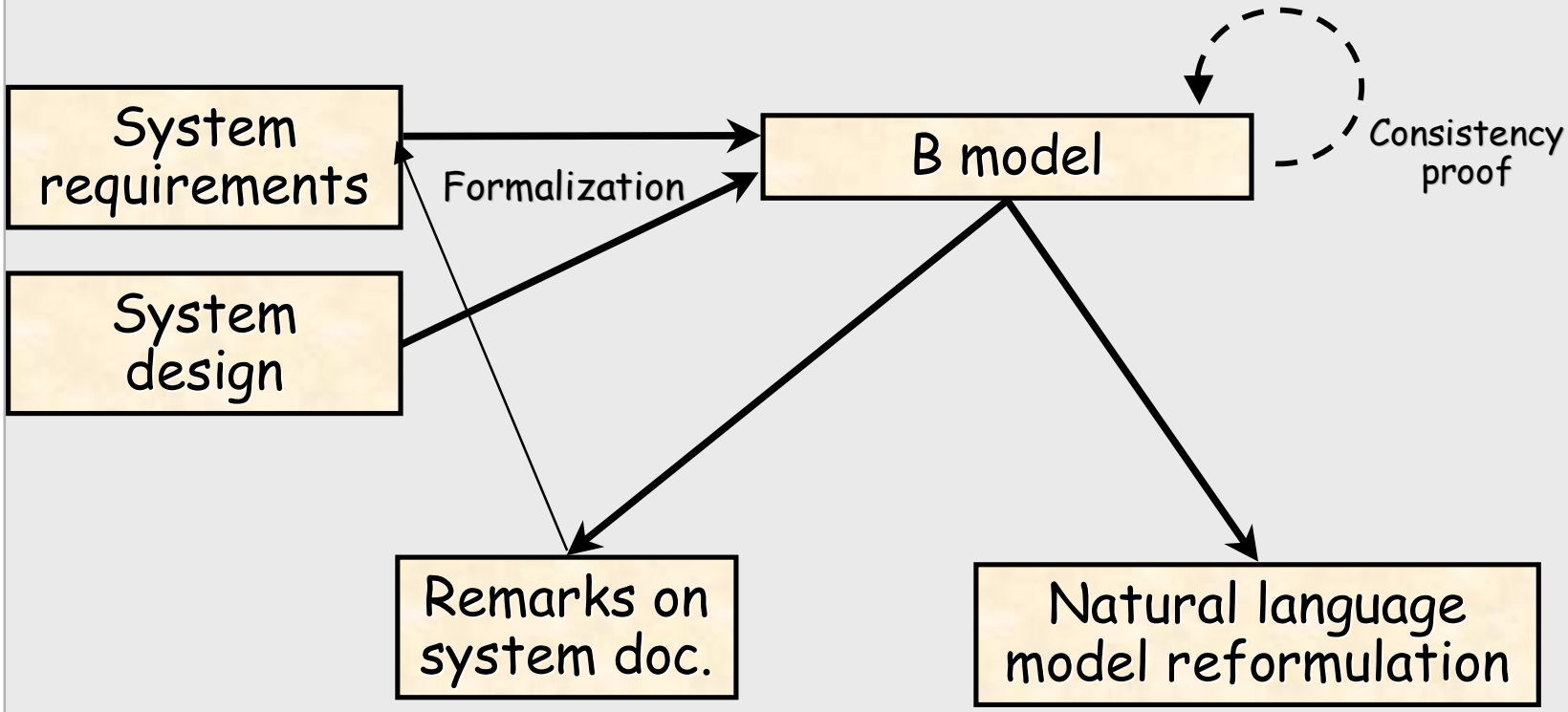
CLEARSY
SYSTEM ENGINEERING

B-Software



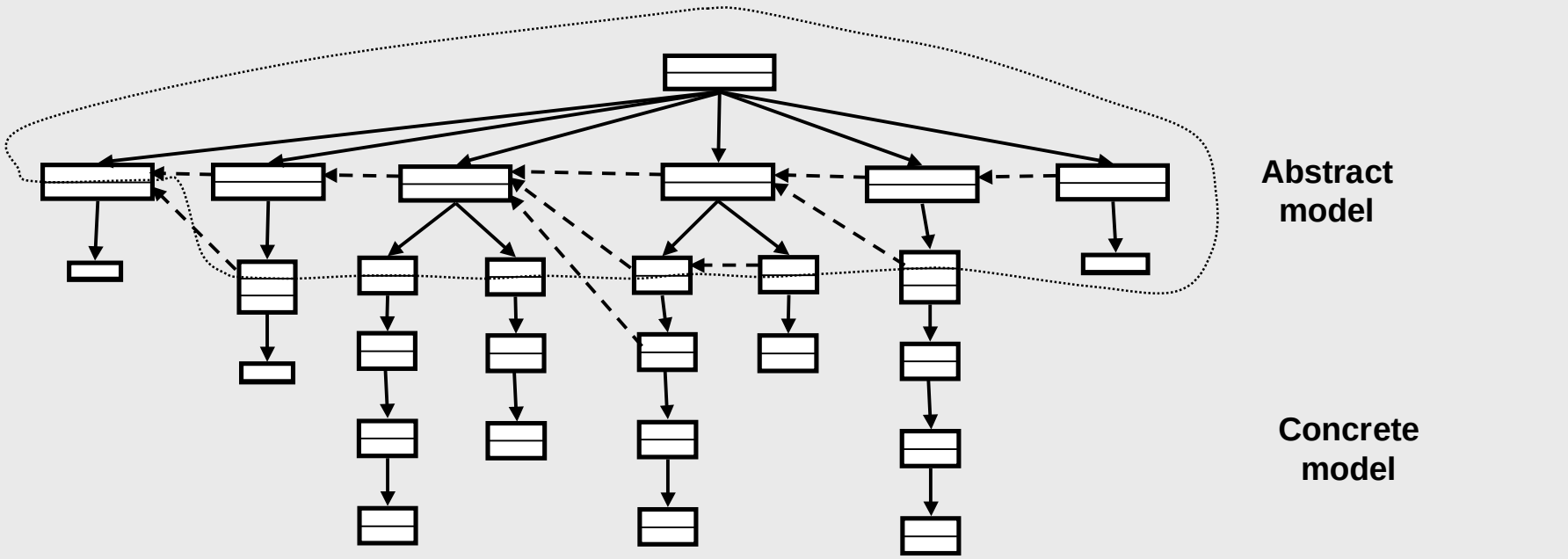
CLEARSY
SYSTEM ENGINEERING

B-System



CLEARSY
SYSTEM ENGINEERING

B-Software structure



CLEARSY
SYSTEM ENGINEERING

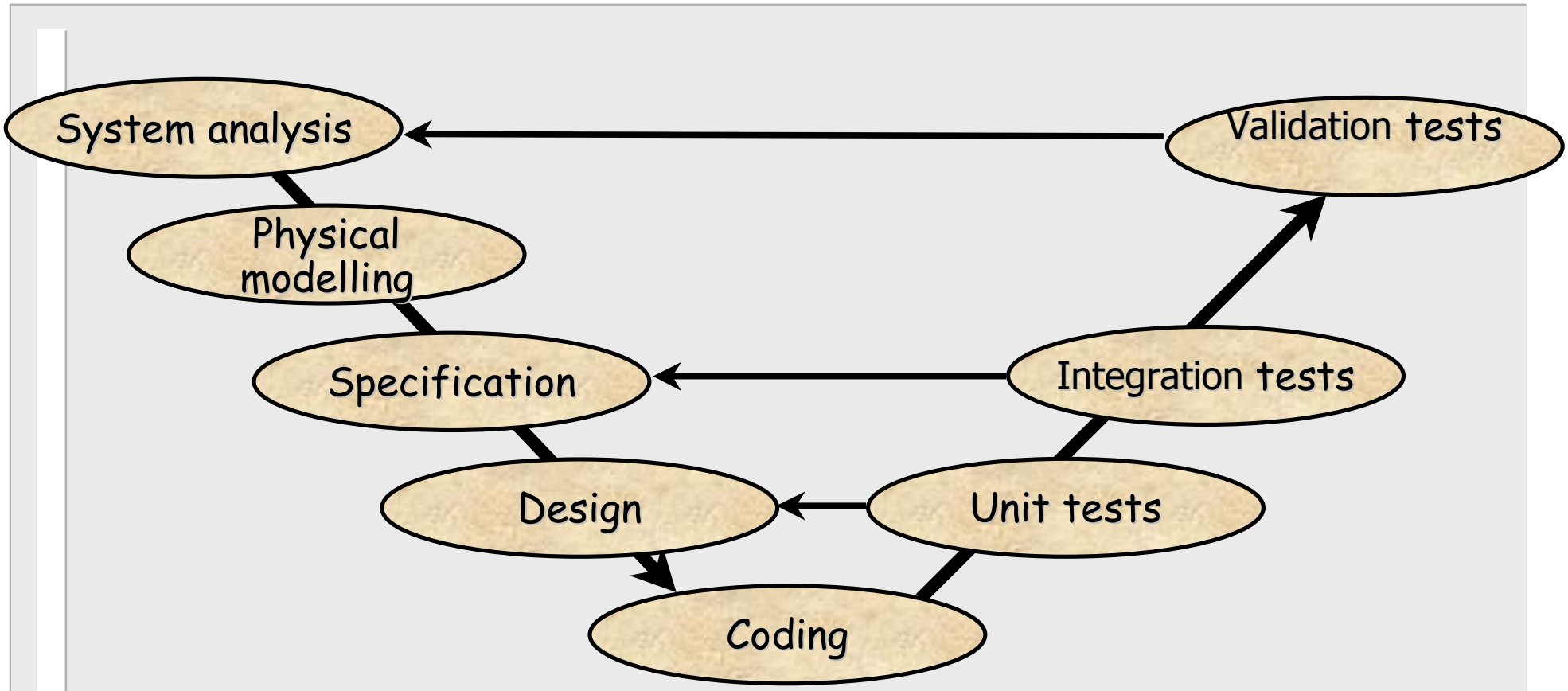
benefits of B-Software

- *The Abstract Model*
  - Requirements are formalized into B specifications module by module
    Non-formal and formal specification are very close (they both express **what** the software should do) to minimize errors
  - Some software properties are formalized into B
    They strengthen the B model, since we must **prove** that they remain true when the modules are put together
- *The Concrete Model*
  - We must **prove** that the concrete model complies with its specification (the abstract model)

benefits of B-Software

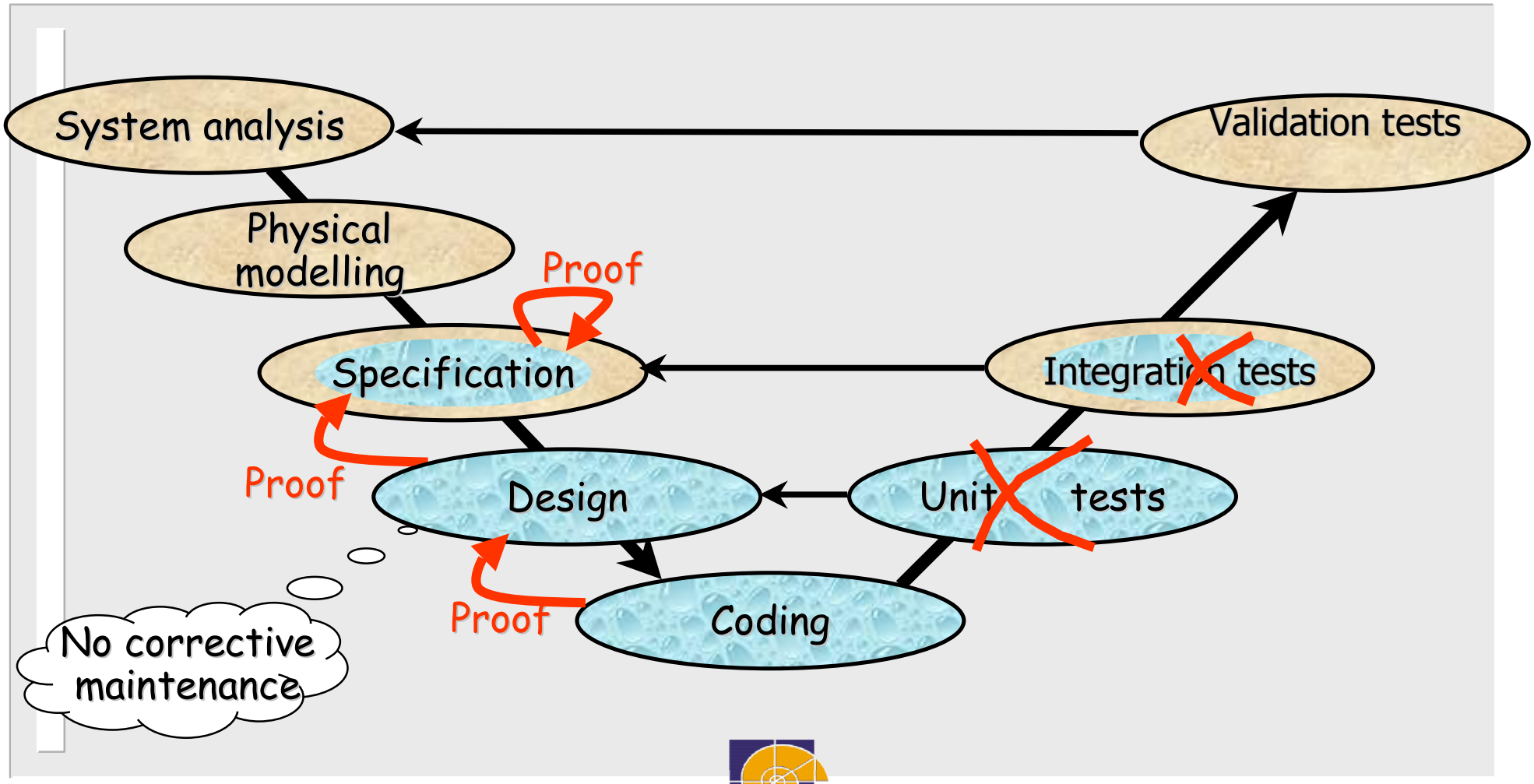
- *The whole Model*
  - NO classic programming error in the code
    (overflow, division by 0, out of range index, infinite loop, aliases)
  - A healthy program architecture
  - Unit Test are no longer used
  - Early detection of errors
  - These benefits remain even after some modifications/evolutions

traditional development cycle



CLEARSY
SYSTEM ENGINEERING

the B development cycle



CLEARSY
SYSTEM ENGINEERING

B compared with programming languages (1)

- *Assembly Language*
  - no static control
- *C Language*
  - limited **static** controls (e.g. typing for data size)
- *Ada Language*
  - extended **static** controls (e.g. strong typing)
- *B Method*
  - static controls + controlling the program **meaning**
    (by **proving** that the B specification is consistent **and**
    by **proving** that the B code complies with its specification)

CLEARSY
SYSTEM ENGINEERING

B compared with programming languages (2)

Ada example

```
-- procedure: maximum
-- input: aa, bb, cc: natural integers
-- output: mm: maximum value
-- description: computes the maximum of 3
--   input parameters and returns it
procedure maximum(aa:NATURAL; bb:NATURAL;
    cc:NATURAL;mm in out: NATURAL) is
begin
    if aa <= bb then
       if bb <= cc then
          mm := cc;
       else
          mm := bb;
       end if;
    else
       if aa <= cc then
          mm := cc;
       else
          mm := aa;
       end if;
    end if;
end maximum;
```

B example

```
mm <-- maximum(aa,bb,cc) = PRE
    aa: NAT & bb: NAT & cc: NAT
    THEN
       mm := max({aa,bb,cc})
    END

mm <-- maximum(aa,bb,cc) =
BEGIN
    IF aa <= bb THEN
       IF bb <= cc THEN
          mm := cc
       ELSE
          mm := bb
       END
    ELSE
       IF aa <= cc THEN
          mm := cc
       ELSE
          mm := aa
       END
    END
END
```

CLEARSY
SYSTEM ENGINEERING

benefits of B-System

- *B-System model*
  - The bottom line is to deeply understand the system through the B model construction
  - A work in cooperation with system creators
  - Early detections of errors, at system design level, producing better Software Specification
- *Remarks on the system*
  - most questions on the system arise during creation of the B model
  - a few inconsistencies may also be detected through model proof (since the model should be consistent in every possible case)
- *Produces*
  - interesting remarks on the system
  - a natural language reformulation of the model giving a sharp, concise and highly structured system description

The Tools (1)

- *Atelier B (ClearSy, www.clearsy.com)*
  - created to develop industrial B-Software projects
  - a set of tools integrated into a project manager tool
    - static checker
    - automatic proof obligation generator
    - automatic provers and interactive prover
    - code translators: Ada, C, …
  - it is also used for B-System
- *B4free (www.b4free.com)*
  - free but restricted to academic users and owners of Atelier B
  - the core tools of Atelier B + a new xemacs interface
- *Rodin platform (September 2007)*
  - a new open platform dedicated to B-System (under construction)

The Tools (2)

- *Atelier B (ClearSy)*
  - actual version 3.6.5
  - version 3.7 is under construction
    - enhancements dedicated to Software-B developers
    - proving more efficiently
    - a new industrial C translator
  - support (annual fee)
    - mail hotline for using tools / bug correction
    - access to new releases of Atelier B
  - Platforms: Linux, SUN OS, HP UX
    - PC Linux: Atelier B performance increases with PC performance
    - the optimal memory for 1 process is 512MB
    - on large projects: parallel use of several PC
    - Demo version: Linux+AtelierB inside Windows

CLEARSY
SYSTEM ENGINEERING

ClearSy: activities related to B

- development, distribution and support of Atelier B
  - the industrial package includes training sessions
- training sessions for the B Language and Atelier B
  - 4 levels
- a team of consultants to help clients
  - with B-Software / B-System projects

CLEARSY
SYSTEM ENGINEERING

B Training Sessions

www.atelierb.societe.com

- *Level 1: basics*
  - overview of the B method and the B language: model, refinement, proof
  - tutorials and practical with Atelier B: specification, writing a program that is consistent with its specification, notion of proof
- *Level 2: advanced notions*
  - advanced notions of the B method, B for systems
  - tutorials and practical with Atelier B: building a large program, Proof Obligations, B-System study
- *Level 3: proving*
  - learning to use Atelier B to prove a project
  - practical: automatic and interactive proof, proof tools, user rules
- *Level 4: automatic refinement*
  - learning to use automatic refinement tools

ClearSy: activities related to B

- uses B-System to help understand, specify, verify a system development
  - internally (on projects where B is not mandatory)
  - for its clients
- uses B-Software to develop safety-critical software
  - complete developments, from SR Doc to code (CRTC, Roissy)
  - partial developments: from B abstract model to concrete model
  - finish up proof or validate proof of B-software projects
- is part of the B community (EU Projects) and tries to create useful processes and tools based on B
  - Brama: animation of B-System models
  - CompoSys: to provide documentation of B-System models

CLEARSY
SYSTEM ENGINEERING

B in education

- *100 universities/research labs currently active*
- *2000 graduates per year with some experience*

conclusion

- *B is a language*
- *B is a development method*
- *B ensures* correct *systems and software*

- *B is used successfully by industry*
- *B is supported by a tool:* ***Atelier B***
- *B brings concrete benefits to its users*