WP3: Deployment in the space sector

Formal development of the Bepi Colombo pilot

Linas Laibinis, Elena Troubitsyna (ÅAU)

together with

Alexei Iliasov, Alexander Romanovsky (NU)

Åbo Akademi

deploy

Brief history

- Summer 2008 – a large collection of requirements documents
- August 2008 – "white paper" describing modelling approach for Bepi Colombo
- October 2008 – formal models for Bepi Colombo produced by SSF
- November 2008 – analysis of SSF development and a proposal for alternative development

Åbo Akademi

deploy

General challenges

- Formal modelling of service-oriented development
  - Different types of incoming service requests (tele-commands) and outcoming responses (tele-messages)
  - Different tasks: producing scientific data or housekeeping/diagnostic reports, changing execution modes and control flags

Åbo Akademi

deploy

General challenges

- Separate layers for core software and application software
  - The core software (CSW) serves as general interface/middleware
  - The application software (ASW) controls instruments while producing scientific data
  - Incoming tele-commands (TCs) can be targeted to both CSW and ASW

Åbo Akademi

deploy

General challenges

- Modelling execution modes and their changes on different layers
  - Execution mode changes can be requested externally (via special tele-messages) or internally (e.g. after fault detection)
  - Consistency between CSW modes and ASW modes should be guaranteed (no prohibited mode transitions)

Åbo Akademi

deploy

General challenges

- Modelling FDIR (fault detection, isolation, and recovery) mechanisms

- Decomposition into components that can be further developed separately

Åbo Akademi

deploy

Analysis of SSF Bepi Colombo models

- Bepi Colombo models produced by SSF team
  - Very impressive, containing huge amount of useful information
  - Too detailed / concrete for abstract specs
  - Execution of the core and application software is intertwined (problems with decomposition?)

Åbo Akademi

deploy

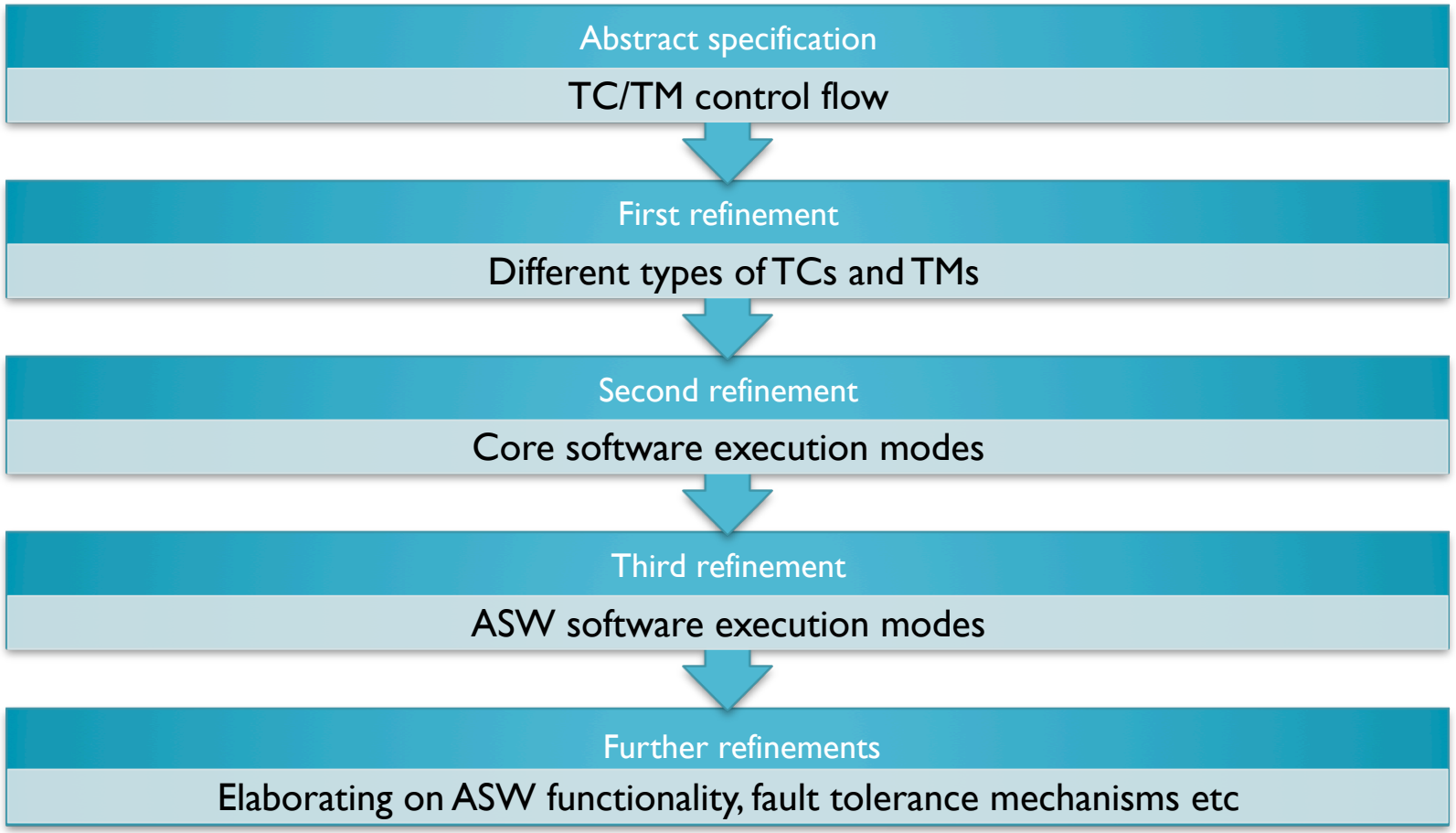
Bepi Colombo models

- To deal with increasing complexity, it would be helpful to have
  - Gradual introduction of concrete data and functionality details
  - More clear separation between CSW and ASW layers

Åbo Akademi

deploy

Formal development by refinement

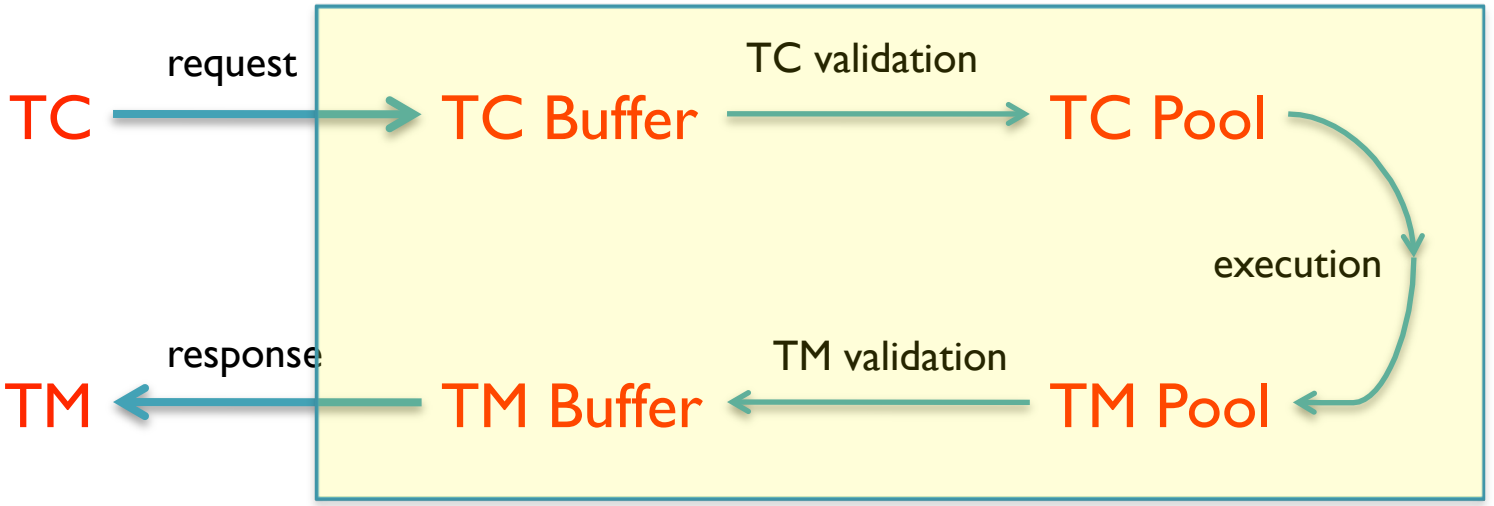


Åbo Akademi

deploy

Abstract specification

- Control flow modelling incoming TCs and outcoming TMs



Åbo Akademi

deploy

First refinement

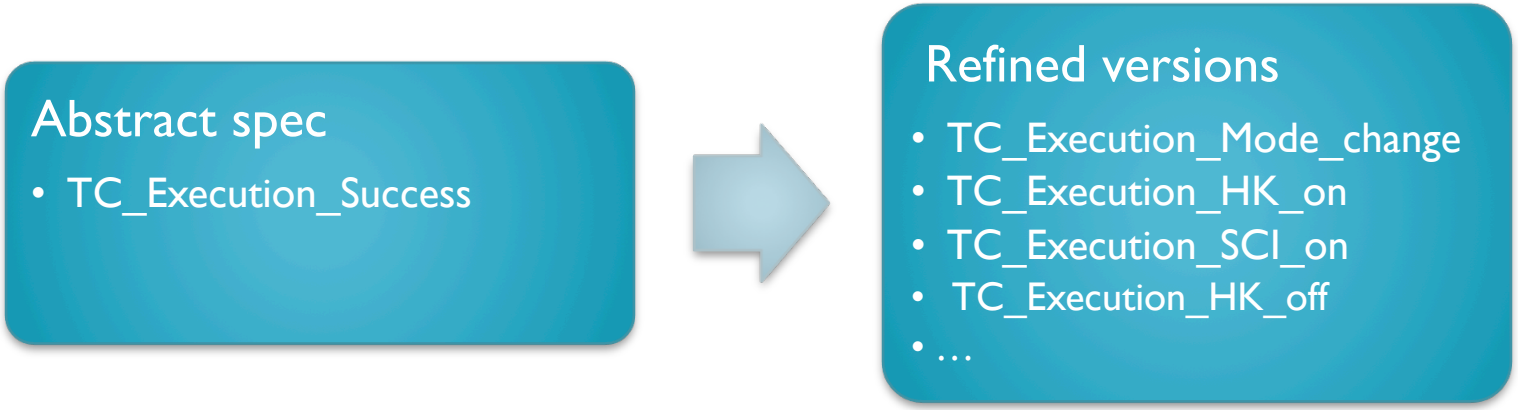
- Modelling different types of incoming TCs and outcoming TMs
  - Abstract data structures for TC/TM types are introduced
  - Their abstract properties (including interdepencies) are defined
  - Concrete instantiation of these data structures is postponed

Åbo Akademi

deploy

First refinement

- Abstract operations are refined, taking into account different types of TCs and TMs



Åbo Akademi

deploy

Second refinement

- Modelling system execution modes and their changes
  - Additional data structures and operations are added
  - Separate cases for TC caused mode changes and FDIR caused mode changes

Åbo Akademi

deploy

Third refinement

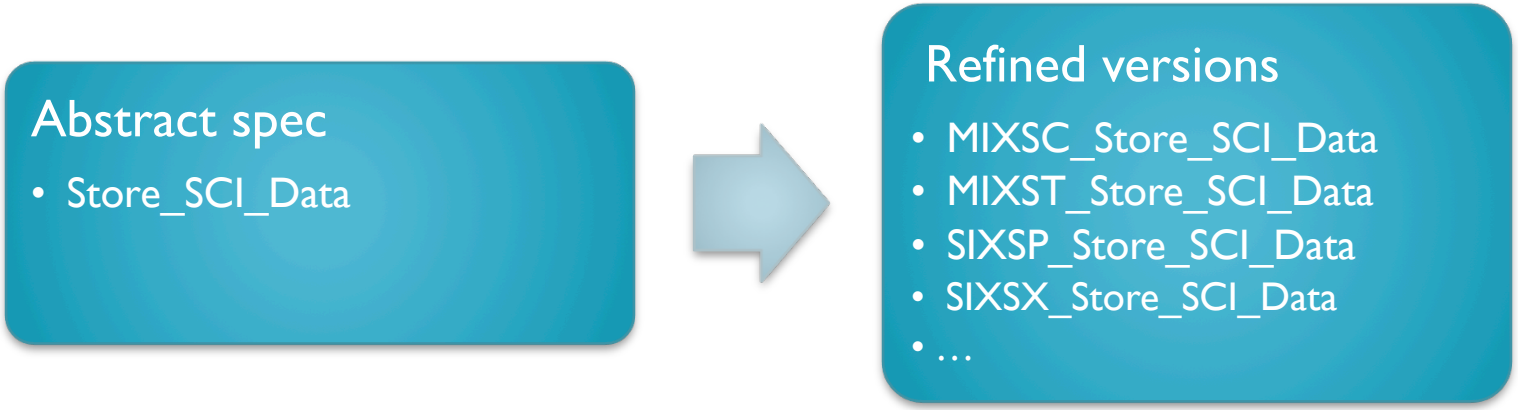
- Modelling instrument (subsystem) execution modes and their changes
  - Additional data structures and operations are added
  - Separate cases for TC caused mode changes and FDIR caused mode changes
  - Interconnection between system and instrument modes is defined
  - Synchronisation (consistency) of mode changes on different levels should be ensured

Åbo Akademi

deploy

Third refinement

- Abstract operations are also refined, creating versions for different instruments



Åbo Akademi

deploy

Further refinements

- As a result of previous refinements, we achieve clear separation between ASW and CSW layers

- We can now focus on ASW subsystems and refine them elaborating on their functionality as well as fault tolerance mechanisms

Åbo Akademi

deploy

Future work

- Reuse of specifications
- Modularisation
- Support for requirements traceability
- Integration into development process

Åbo Akademi

deploy

Thank You!