

Functional analysis of the SRS

Steps

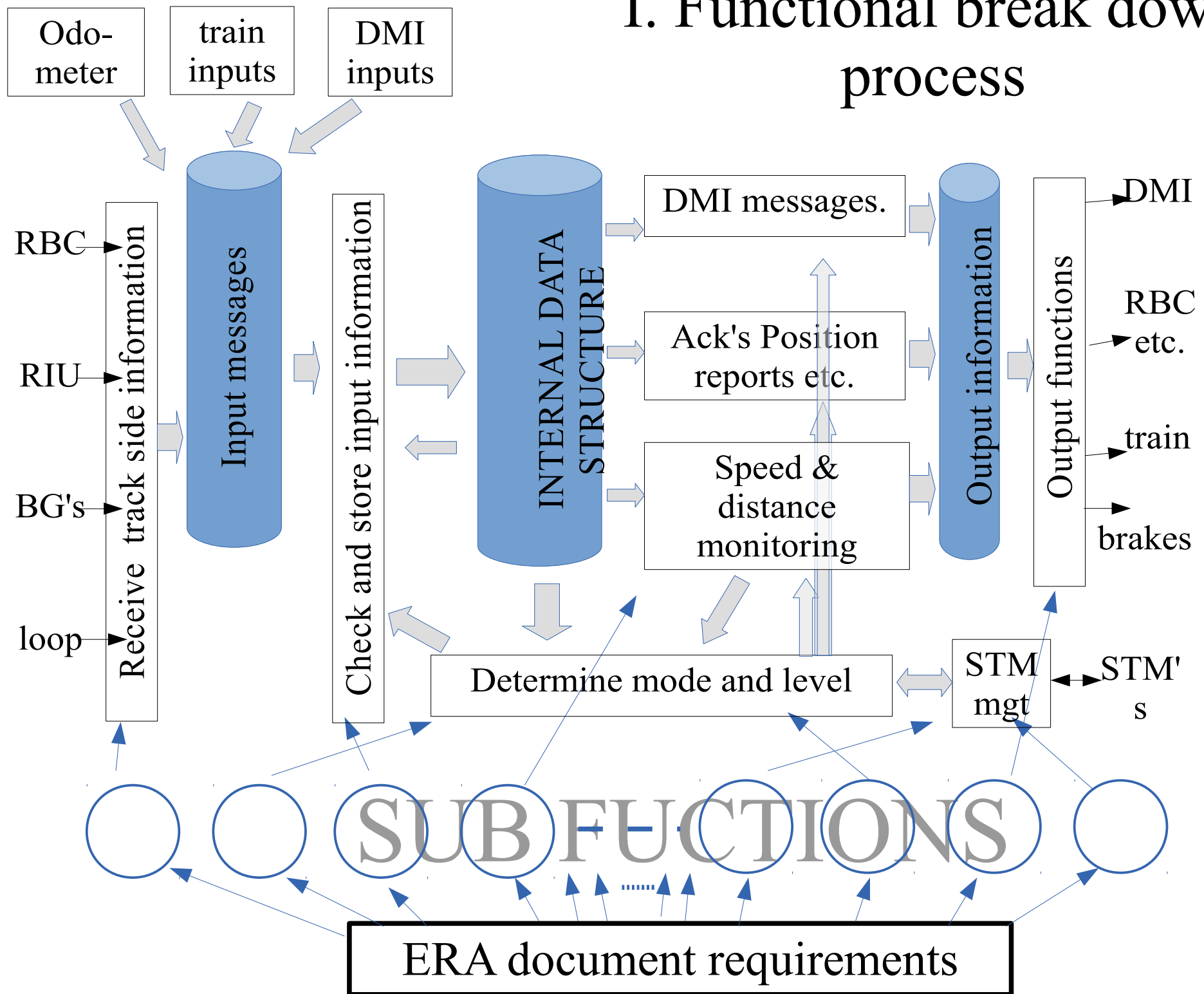
I. Define a high level functional structure



II. Set rules for assigning sub functions to the functional structure

III. Analyse the SRS, resulting in clearly defined sub functions assigned to the functional structure

I. Functional break down process



II. Principles for detailing the architecture

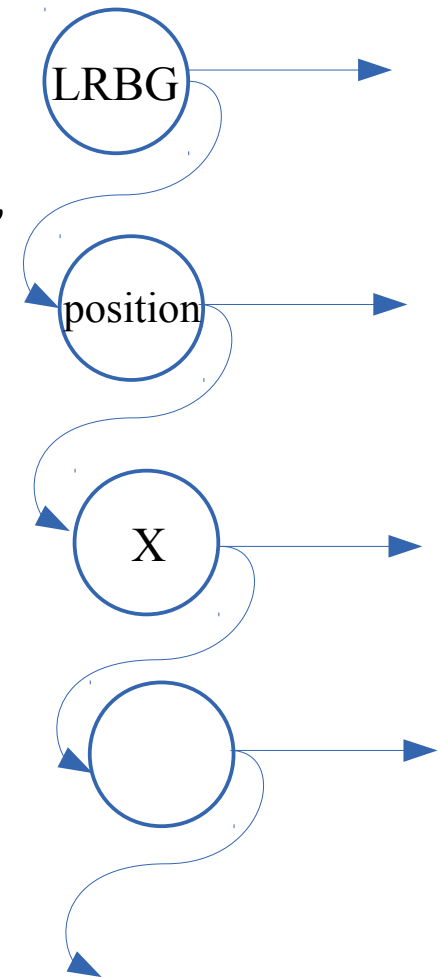
➤ Cyclic execution

- read all inputs received in the current cycle
- update the internal data structure with distance, time, and input (predefined order)
- calculate outputs
- determine mode and level

➤ No external interrupts

➤ Fixed data structure

- internal data stored i.r.t. LRBG (i.e. no incremental distances)



IIIA. Analysis of the SRS

Describe per function:

- Data: in-/outputs
- Requirements with ref's to the SRS (ERA specifications)
- Global description of the functions

Example: format for requirements

Name	Occurrence	Type	Description
Definition	1	T_Definition	Textual and graphical description of the requirement
Nature	1	<ul style="list-style-type: none">• Structural• Functional• Definition	
Source	1	T_SourceDocument	the document where the requirement is defined the first time (SRS, FIS, SSRS,...)
Discussion	1 (Optional)	T_Text	Discussion or comment to clarify or justify the requirement

Discussion: the balance between work and level of formalism

IIIB. Process

Steps in analysing the SRS:

1. Analyse a part of the SRS (e.g. a paragraph)
2. Define the OBU functions (implicitly) described
3. Gather all relevant requirements (also from other parts)
4. Register the coverage of SRS requirements by the defined function
5. Define the in and output variables (also internal)
6. Add the function to the overall functional structure

Requirement structuring

Rule	<i>Negation</i>	<i>Conjunction</i>	<i>Conditional</i>	<i>Disjunction</i>	<i>Existential</i>	<i>Universal</i>
Syntactic relation	Not(...)	(...and...)	(If...then...)	(...or...)	There is x(x is...)	For all x(x is ...)
Semantic evaluation	Not(...) is true iff ⁸ (...) is false. Otherwise it is false.	(...and...) is true iff neither conjunct is false. Otherwise it is false.	(if...then...) is true iff either the antecedent is false, or the consequent is true. Otherwise it is false	(...or...) is true iff either disjunct is true. Otherwise it is false	There is an x (x is...) is true iff there is an element of the universe of discourse such that the entity satisfies the properties ascribed to x.	For all x (x is ...) is true iff every possible individual in the universe of discourse satisfies the properties ascribed to x.

IIIC. Example MA

- MA's can be given using 3 different data structures
- Distances given incremental
- Maximum number of sections not clear
-

Data structure and sub functions defined:

Sub functions for storing and updating an MA

- Initialize the MA according to the default values.
- Check if a received MA is valid (direction, ssp, gradient available,....)
- Delete the existing MA (as far as necessary) if a new MA is received.
- Store MA data (except timers)
- Store and update timers when a new MA is received
- Update MA for time elapsed: Check timers and apply applicable actions
- Update MA for a new LRBG
- Update MA for a distance driven

Example data structure MA

8.3_1

MA structure: (one instance of this type only)

The number of sections in the MA (default 0, maximum including the end section: 34)

Sections as defined in 8.3_3

End section information (additional information concerning the furthest MA section):

- An “end section timer start location” (related to the LRBG) (default: infinite)
- An “end section timer” (see 8.3_4)
- Limit of authority speed. (default: 0)
- A Limit of authority “(LOA) timer (see 8.3_2)
- Danger point distance: distance beyond EOA to the danger point (“0” if not available) (default: 0)
- The release speed related to the danger point. (default: 0)
- Overlap distance: the distance beyond the EOA to the end of the overlap (default: 0)
- The release speed related to the overlap. (default: 0)
- The overlap timer start location (related to the LRBG) (default: infinite)
- An “overlap timer” (see 8.3_4)

The signalling related maximum speed (only applicable in level 1)