EN 50128 REQUIREMENTS
FUNCTION BLOCK DIAGRAM (FBD) PROGRAMMING

Safety Manager Janne Peltonen, MIPRO Oy

TOPICS

- ✓ MIPRO Oy – Finnish system integrator
- ✓ Advantages of previously certified and proven COTS Safety PLC platforms
- ✓ EN 50128 requirements for LVL application programming
- ✓ Safety management
- ✓ Development and testing aspects

MIPRO in a nutshell

- Independent SYSTEM INTEGRATOR since 1980
  - automation and information systems
  - staff of 55 professionals
  - Main office in Mikkeli, Finland
  - Branch offices in Oulu and Helsinki

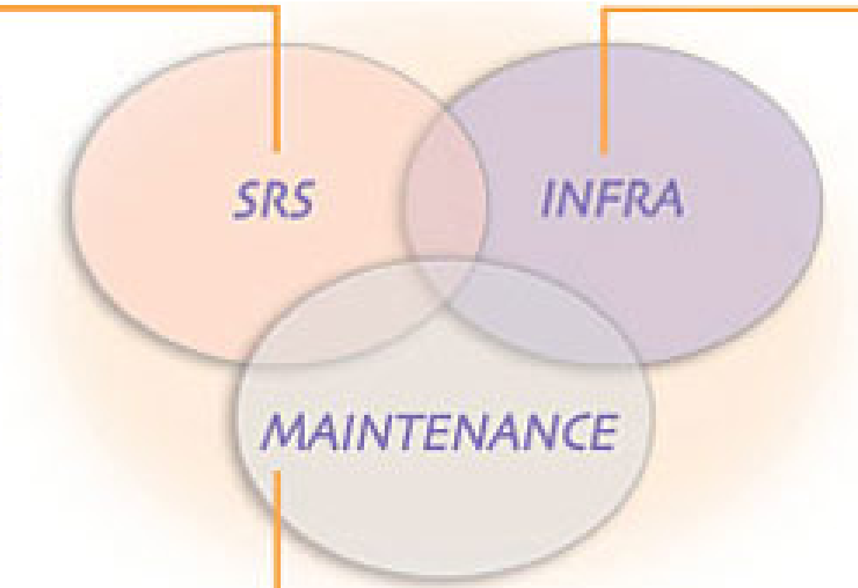
- Main business area SAFETY-RELATED SYSTEMS
  - Railway signalling systems and level crossings
  - ESD systems and machinery safety systems for industries
  - Central Train Control (CTC) systems for railways

- Main business area INFRA
  - Waterworks, wastewater treatment, boiler plants etc.

MIPRO

specialized

Safety Related Systems for Railways and industry



Automation for Water and Wastewater Works

Maintenance and after-sales services

Long Term Partnership History

- **Infrastructure Customers**
  - **Continuous services and revamps**

- **Industry**
  - **Chemical Plants and Boiler Plants**
  - **Metal Manufacturing**
  - **Mining**

- **Finnish Railways**
  - **Over 10 years of continuous development and co-operation**
  - **Frame Contract to supply Safety Related Control Systems for Traffic Control**
  - **Maintenance Support Contract**

Safety Related Systems

- Expertise in programmable safety systems
  - safety-critical system deliveries since 1987
  - 30+ professionals in safety-related projects
  - 6 TÜV-certified safety engineers
  - software development
  - main representative of HIMA in Finland
  - consulting services
- Adoption of international safety standards
  - ISO 9001 quality management system
  - IEC 61508 product certification
  - IEC 61511 process industries
  - IEC 62061 safety of machinery
  - EN 50126, EN 50128, EN 50129 railways

MIPRO on Tracks

- Development for railways since 1990
- First Level Crossing Control commissioned 1995, still going strong
  - Installed base of 50+ mainline Level Crossing Controls - and some others
- Interlocking development started 1997
  - Commercial, well known Safety PLC with excellent tools makes a trusted platform
  - Field proven SCADA has all the necessary basic functions and the continuity
- First Interlocking commissioned 1998
- MiSO TCS Safety Case presented to RHK (EN 50129), 2001
  - Quality System
  - International Standards

1 / 2

MIPRO on Tracks

- **After international competitive bidding 2002, MIPRO was appointed to build most of the ”ATP 3rd phase, 2002-2006” Interlocking systems**
  - **2400+ rail kilometers covered already**
  - **130 + systems commissioned**

- **Mipro was appointed to build the Oulu CTC, 2003**
  - **30 000 + active Database objects, 50+ Interlocking Systems**

- **International subcontracting for ANSALDO, 2003**
  - **Jyväskylä-Pieksämäki line, MiSO Remote/CTC, System Installation and Wayside contracting**

- **5 year contract for Ilmala Interlocking, 2007**
  - **Main service depot in Finland (60ha area, 55km of tracks, 260+ points)**

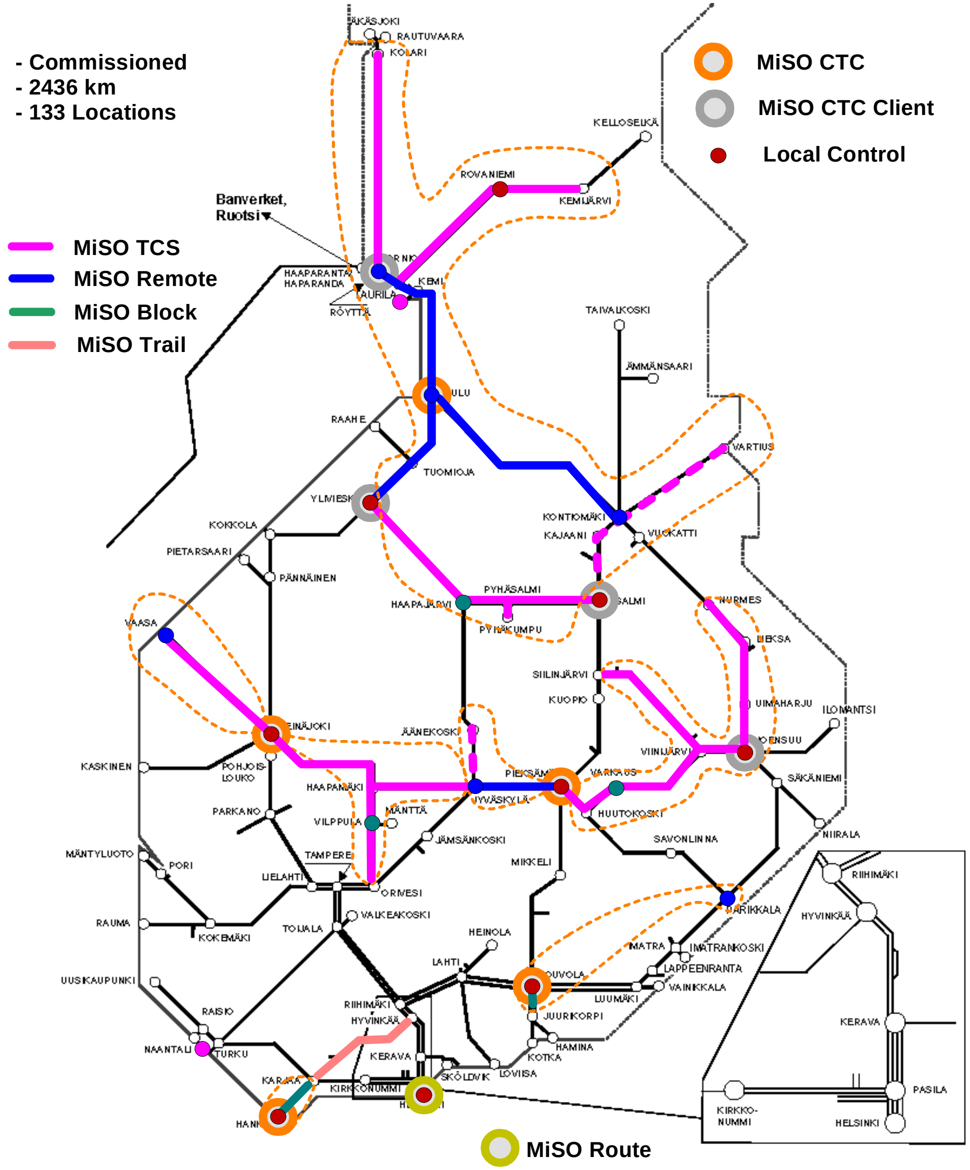
MiSO Systems 1998 - 2006

Oulu CTC (extended functions) 2003-2005
- Oulu, MiSO Remote, 2003
- Oulu -Tornio, MiSO Remote, 2004
- Oulu - Kontiomäki, MiSO Remote, 2004
- Oulu - Ylivieska, MiSO Remote, 2003
- Tornio - Kolari, MiSO TCS, 2003
- Laurila - Kemijärvi, MiSO TCS, 2004
- Iisalmi -Ylivieska, MiSO TCS, 2004
- Iisalmi - Kontiomäki - Vartius, MiSO TCS, commissioning 2006
- TrainNumber, DispatchAutomation, MiSO Graphics, 2005

Pieksämäki CTC (basic functions), 2003
- Pieksämäki - Joensuu; Siilinjärvi – Viinijärvi, MiSO TCS, 2003
- Joensuu - Nurmes, MiSO TCS, 2005
- Parikkala (- Savonlinna), MiSO TCS, 2005, 2006
- Jyväskylä - Pieksämäki, MiSO Remote, commissioning 2005
- Jyväskylä - Äänekoski, MiSO TCS, commissioning 2005
- Jyväskylä, MiSO Remote, commissioning 2005

Seinäjoki CTC (basic functions), 1998, 2004
- Orivesi - Haapamäki - Jyväskylä, MiSO TCS, 2003
- Haapamäki - Seinäjoki, 1998, 1999
- Seinäjoki - Vaasa, TCS, 2001

Other systems
- Karjaa - Hanko, MiSO Block, 2000
- Tornio - Kolari, MiSO Trail 1997, decommissioned
- Hyvinkää - Karjaa, MiSO Trail, 1998
- Helsinki Interlocking, MiSO Route, 2001
- Helsinki Interlocking, Monitoring, 2003
- Lappeenranta - Parikkala, MiSO Line , 2001, decommissioned
- Kouvola - Inkeroinen, MiSO Block, 2005
- Level Crossing Control Systems, 1995-



EN 50128 requirements for software

- Requirements for quality and safety management
- Requirements for software functionality
- Requirements for software safety integrity
  - Software has only systematic failures
- Requirements for software verification and validation
  - Everything needs to be checked, tested, assessed and approved
- Requirements for software configuration management
- Requirements apply to several software lifecycles
  - Hardware level embedded software development
  - Application level software development

Safety Integrity Levels (IEC 61508)

Safety integrity level	**Low Demand Mode of Operation** (Average probability of failure to perform its design function on demand)	**High Demand or Continuous Mode of Operation** (Probability of a hazardous failure per hour)
4	$\geq 10^{-5}$ to $< 10^{-4}$	$\geq 10^{-9}$ to $< 10^{-8}$
3	$\geq 10^{-4}$ to $< 10^{-3}$	$\geq 10^{-8}$ to $< 10^{-7}$
2	$\geq 10^{-3}$ to $< 10^{-2}$	$\geq 10^{-7}$ to $< 10^{-6}$
1	$\geq 10^{-2}$ to $< 10^{-1}$	$\geq 10^{-6}$ to $< 10^{-5}$

Clear definition – Unclear demonstration

How to calculate probability of failure for SOFTWARE?

EN 50128 requirements - demonstration

- System Safety Plan
- Hazard Log
- Software Quality Assurance Plan
  - Records of quality and safety management activities
  - Competency, responsibilities and independence of personnel

EN 50128 requirements - demonstration

- Software Functional Safety Requirements Specification
- Software Interface Requirements Specification
- Software Architecture Specification
- Software Safety Integrity Requirements Specification
    - Adequate methods according to EN 50128
    - Software safety integrity may differ from hardware safety integrity

EN 50128 requirements - demonstration

- Software Verification and Validation plan
  - Audit, review and inspection records
  - Test and analysis records

- Software Configuration Management plan
  - Identifiable and traceable record of approved software
  - Identifiable and traceable record of compatible hardware platform

- Safety Case for Independent safety assessment
  - Safety Assessor from Independent organisation
  - Assessor is not responsible for any testing activities and requires clear and auditable documentation

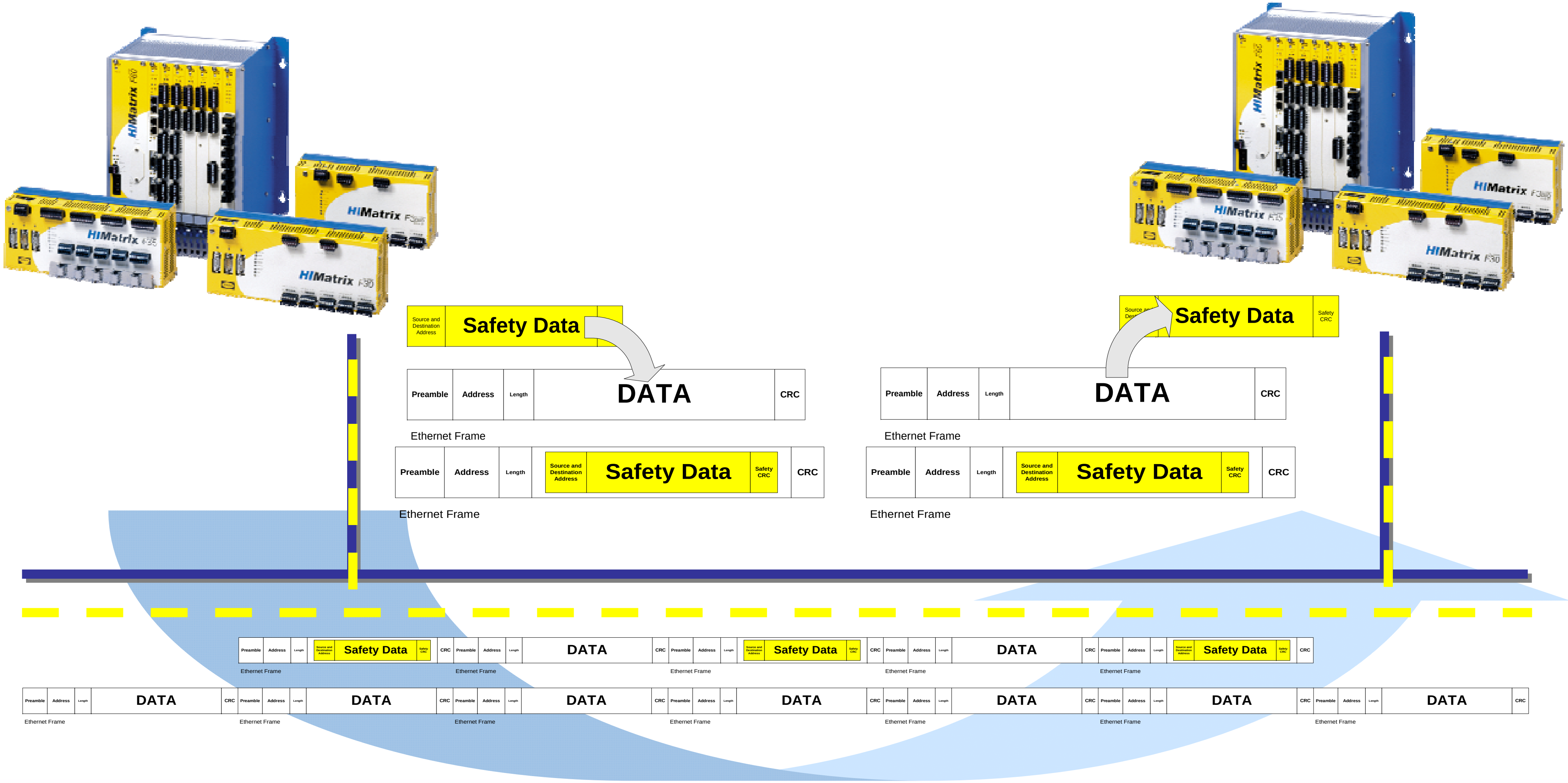
SIGNALLING SYSTEM LOGIC PART

- Safety certified (TÜV) COTS Safety PLC
- Extensive field experience
  - Installations and agents all over the world
  - Chemical industry
  - Manufacturing industry
  - Mining industry
  - Energy industry
- Certified and tested Generic Product platform for signalling system

LOGIC PART EMBEDDED SOFTWARE

- Safety certified (TÜV) embedded software
  - Code generator
  - Central module OS
  - Data communication OS
  - Ethernet-based safety-critical communication protocol
- Safety bus communication protocol
  - A must for railway applications
  - Safety guaranteed with EN 50159 approach
- Application software code generation from FBD
  - Only application part of software needs verification

SafeEthernet concept



SafeEthernet – Error detection methods

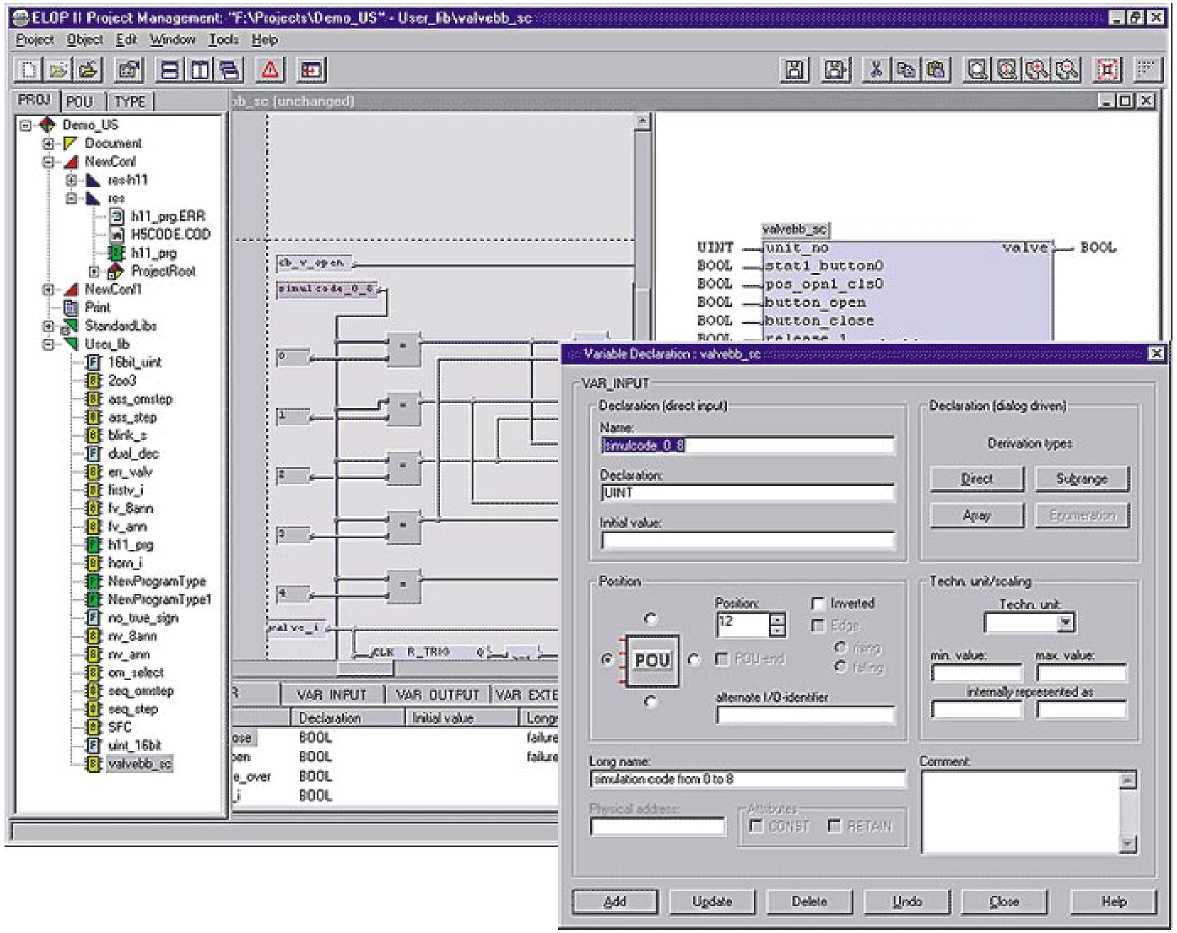
Error	Methods: sequential number	timestamp	notice of receipt	identific. of sender and receiver	data backup	redundancy with cross-comparrison
Iteration	X	X				X
Loss	X		X			X
Insertion	X		X	X		X
Wrong Order	X	X				X
Message corruption			X		X	
Delay		X				
Connecting Safe and Non-Safe			X	X		

PROVEN-IN-USE SUPPORT TOOLS

- IEC 61131-3 compliant FBD programming
  - Programming environment minimizes verification tasks
  - Programming environment minimizes human errors
  - Visual verification is possible

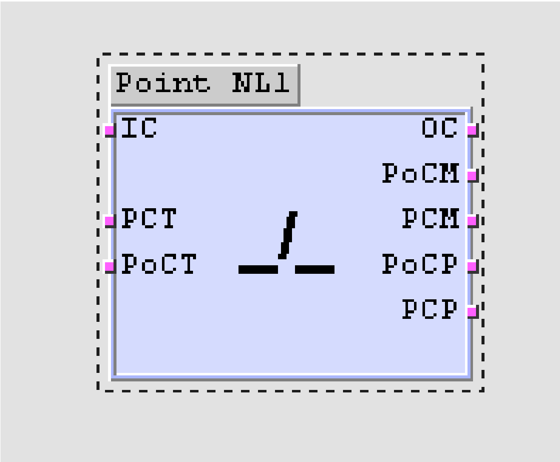
- Centralized maintenance supervision
  - Access to all diagnostics information
  - Program state monitoring online
  - Remote maintenance support

Programming environment



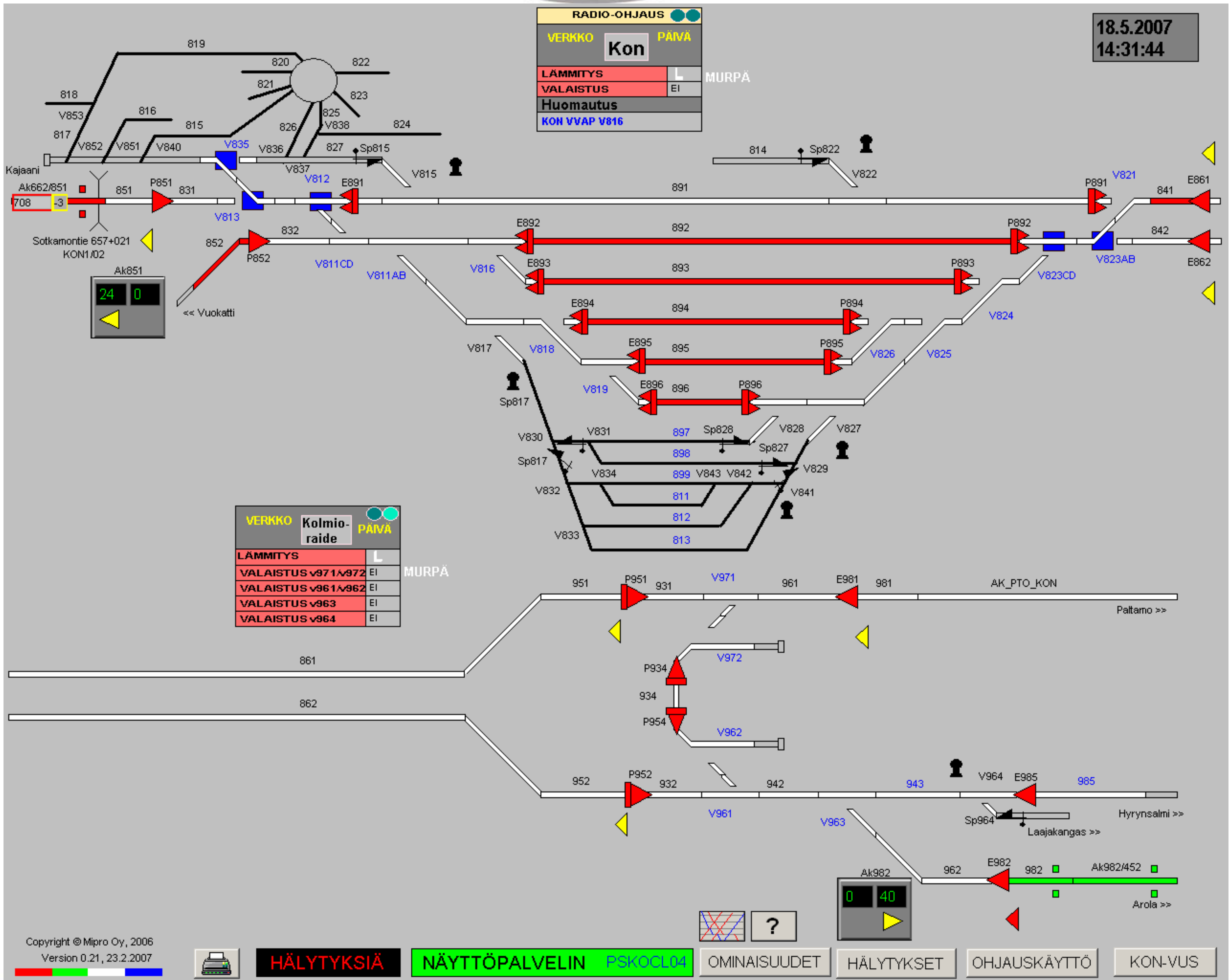
- Drag & drop
- Automatic consistency checks
- Checking tools
- Import/export functions

Application software module



- Full safety testing during development
- Formal interface minimizes human errors
- Storage of application related know-how
- Increased confidence through operation years
- Re-use allows rapid project implementation

Train Dispatcher’s World



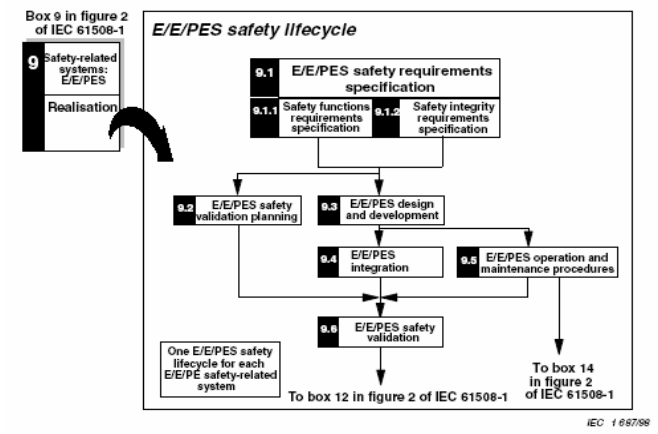
Sound expertise in customised automation

SIGNALLING SYSTEM HMI

- Safety requirements SIL X?
- Practical safety issues to consider
    - Critical commands and confirmation
    - Indications and decision making
    - Alarm handling and diagnostics
    - Integration of different signalling systems
    - Minimizing choices
    - Automatic route control and wide-area remote control
- New generation of train dispatchers
    – Training, instructions and online help

Functional Safety Management

- ❑ Mandatory for safety related projects
- ❑ Requirements of safety standards
  - – Evidence of quality management
  - – Evidence of safety management
- ❑ Definition of…
  - – Business processes
  - – System safety lifecycle

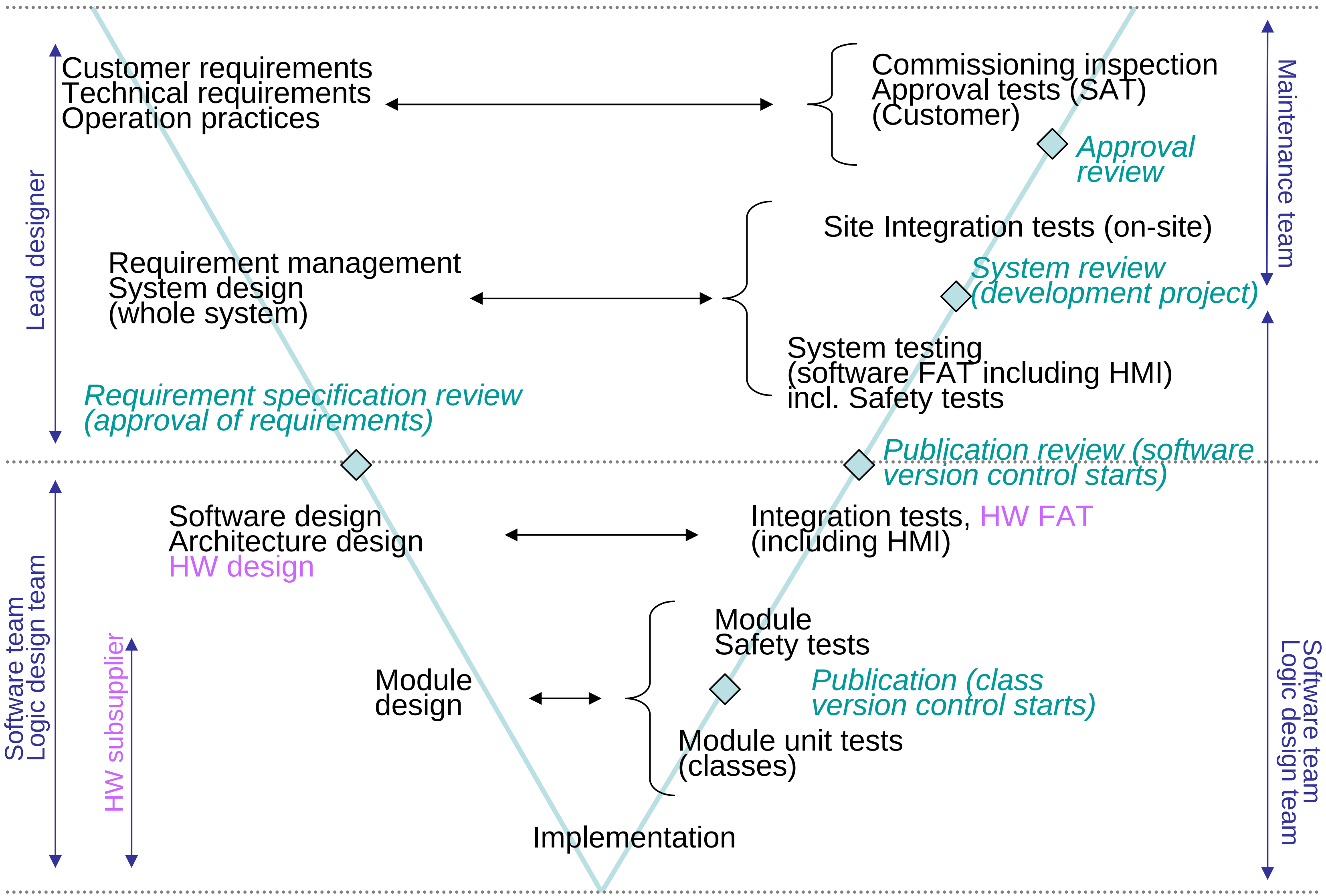


- ❑ Applied basic requirement : single human error may not cause loss of safety function !

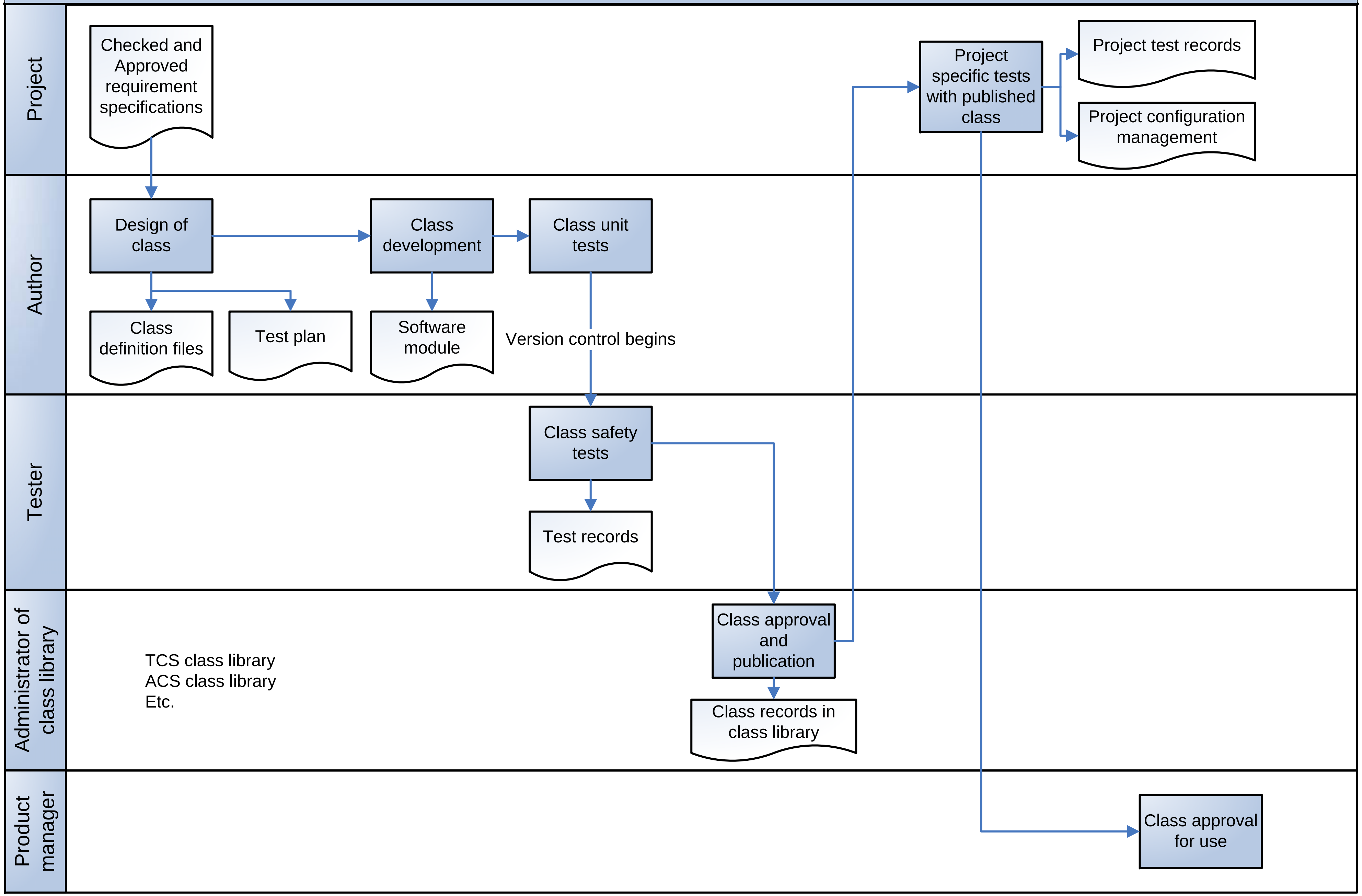
Functional Safety Management

- Project Safety Plan
  - Organizations, responsibilities & competencies
  - Hazard and risk analysis
  - Definition of system requirements and interfaces
  - Verification & Validation activities
  - Documentation & flow of information
  - Use of previously approved products and solutions
  - Independent assessment prior to start-up
- Risk from poor safety planning may be greater than risk from individual equipment

Software development V-model (MiSO application software)



Management of application software classes (modules)



Examples of practices

- Testability and understandability
  - Standard and classified variable names
  - Test planning covers entire functionality and any thinkable discontinuity and fault conditions
  - Safety functions are separated from other functions
  - Program state can be monitored from outside
  - Program records the first failure causing the stop
  - Program modules exchange information only through external visible interfaces
  - Freely programmed part of modules is defined and described
  - Commentation of program and explanation of restrictions
  - References to requirements specification inside program
  - Monitoring with automatically generated logs

Examples of practices

- Minimizing time-dependent characteristics
    - Avoidance of delays and pulses
    - Program execution monitoring
    - No parallel execution paths
    - Application of state machine design
    - Time windows for functions
    - Consideration of data communication delays
    - Monitoring/filtering of field data change rate

Examples of practices

- Verification of safety-critical information
  - Alarming signal range errors
  - Monitoring of data communication
  - Announcement of critical commands to operator
  - Use of combined signals instead of single signals

- Program identification and version control
  - Version control and modification management
  - Verification and Validation
  - Version identifiers inside the program

THANK YOU

FOR MORE INFORMATION:

janne.peltonen@mipro.fi

www.mipro.fi