IRONMAN

PTN

Support For Graphical Clients M10

Date:	31/03/2016 16:43:00
Version:	V0.1
State:	**in work**
File name:	GUI-support.doc
Number of pages:	89
Authors:	B.Döring, D.Razumov (SBS)

Отформатировано: английский (Великобритания)

Contents

Отформатировано: английский (Великобритания)

SIEMENS

Ironman

CHAPTER 0 Overview

Document History

Date	Version	
2005-10-07	V0.1	Initial Version, just for internal discussions

Purpose

This document discusses topics related to concepts and architecture for graphical clients to be connected to Ironman. Due to the work split between StepLogic and Siemens Business Services this document focus on clients using graphical drawings like network maps (and schemes???).

Scope

PTN, Release 2 and/or 3

Definitions

- Business Object: Interface objects between clients and Ironman Server.
- Graphic Object: Objects modeling graphical data like icon, size, color and position of graphical representation of TMN data.
- Managed Object: Objects modeling TMN data (and administrative data like Ironman users); stored in Ironman database

Abbreviations

BO	Business Object
GO	Graphic Object
MO	Managed Objects

References

/1/
<title>
<version/date>

Отформатировано: английский (Великобритания)

Alias Names

-

Remarks

This document version is just for internal discussion and collects ideas. Later version will specify the final concept.

Отформатировано: английский (Великобритания)

CHAPTER 1 Context

This chapter is a copy from PTN-GUI M30 documentation.

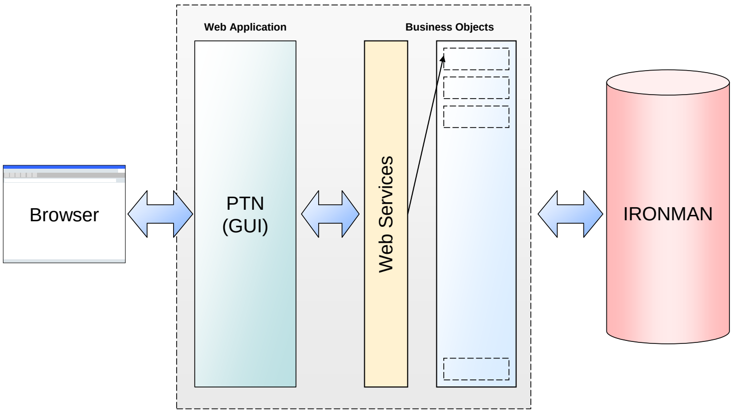
1.1 Basics

The proposed GUI for IRONMAN@PTN Release1 will be realized as a WEB-Application running in a WEB-Platform (Apache,Tomcat,AXIS,…) and will be accessible via a WEB-Browser (i.e. Internet-Explorer).

For the visualization of templates & journals/lists the standard HTML-Features/-Controls will be used (i.e. INPUT, TABLE,…). For the visualization of graphical parts (scheme, drawing,…) the graphic format SVG will be used, which can be presented inside a WEB-Page using a "SVG-Viewer"-Plugin.

For the access to the IRONMAN Object-Data, the IRONMAN BusinessObject-Plattform will be used, which allows clients to access the IRONMAN-System via a WebService.

1.2 Architecture Overview



Отформатировано: английский (Великобритания)

1.3 Platform

- Browser: Microsoft InternetExplorer >= V6.0 (+ JavaScript enabled)
- SVG-Viewer: Adobe SVG-Plugin >= V3.0

1.4 List of Journals, Templates, Passports and Drawing Schemes

In general there are five different types of the User Interface presentations:

1) **Journal**: list of a selected set of objects
2) **Template**: input and edit mask of the data of single or several objects
3) **Passport**: report according GOST standard
4) Drawing **Scheme**: report including logical graphical representation according GOST standard
5) Graphic Drawings: drawings with location information according GOST standard

1.5 General functionality

Примечание [A1]: I think this list must be in this document

- Select an object on a Journal and open the object dialog
- Select an object on a Journal and put its reference information to an attribute of another object dialog (e.g. Manhole-A, Manhole-B in a Cable Canalization)
- Enter an object identifier directly and open the object dialog
- Sorting of entries in a Journal by columns (alphanumerical, 1 column sorting criteria)
- Hide/visible selected columns in table
- Open an object dialog for creation of an new object
- Select an object dialog (e.g. from Menu)
- Select a function on an object (e.g. from object Menu)
- Of course edit attributes
- Whenever possible give value lists for attributes, so that the user must not type in the attributes
- Mark mandatory attributes
- Protect attributes which are not modifiable
- Fill in attributes of other objects automatically if they have to be shown in another object dialog
- Print passports and schemes (parts of schemes of cable canalization too)
- Scrolling vertical and horizontal in schemes, tables and on „maps“:
  tables: the side is scrolled, not the table;
  schemes: SVG-behavior: key + mouse, drag via overview
- Avoid horizontal scrolling in tables whenever possible (provided by hide-column functionality)
- Error messages have to be linked to the attribute which has to be corrected if possible:
  => depends on the messages from OS, BO’s
- Error messages should have an error code for diagnostics
- Zoom on maps, schemes (cable canalization, description of cables in buildings)

Отформатировано: английский (Великобритания)

CHAPTER 2 Topics to Be Considered

2.1 Overview

This chapter shall be used as a check list. It does neither contain discussions nor solutions.

2.2 List of Important Topics

The following shall serve as a check list. If we come later to the conclusion, that some topic is not relevant for the concept and the architecture, this topic will not be worked out.

ID	Type	Description
	Concept / Architecture	Are there common constraints, which have to be considered? Subtopics: • Basic architecture rules (design pattern, good practice, …) • Constraints out of existing solutions • Constraints out of selected products
	Architecture	Which subsystems/modules are defined? Only those subsystems/modules have to be considered here, which interact with subsystems/modules of different suppliers.
	Architecture	Which subsystem/module is responsible for which behaviour?
	Architecture / Data Model	Which subsystem/module is responsible for which data?
	Architecture	Which subsystems/modules interact with each other? Subtopics: • How subsystems access data of other subsystems? • Which subsystem is client and which one server (in general or in special use cases).
	Architecture	Which subsystems/modules are defined? Only those subsystems/modules have to be considered here, which interact with subsystems/modules of different suppliers.
	Interface / Communication	Which kinds of protocols and data formats are used at the interfaces?
	Interface / Communication	Which kinds of protocols and data formats are used at the interfaces? Subtopics:

Отформатировано: английский (Великобритания)

ID	Type	Description
		• Synchronous or asynchronous communication? • Notifications? • Basic rules of data formats (e.g. common rules, how xml data structures are build). At this level we do not need detailed description of each message/data structure.
	Architecture	Which subsystems/modules interact with each other? Subtopics: • How subsystems access data of other subsystems? • Which subsystem is client and which one server (in general or in special use cases). • Which kind of data access is allowed at which interfaces (read, create, modify, delete) in principle.
	Architecture / Security	Single sign logon? Which subsystem is responsible for which security topic? Subtopics: • Password check • Access rights • Data encryption
	Architecture	Which software runs on which machines? Which subsystems/modules are linked together to program units? Which software is deployed where and in which environment?
	Architecture / Naming	Which naming is used to identify objects in other subsystems/modules?
	Architecture / GUI Design	Style guide to be used? How to implement a common design and behaviour, although parts are implemented by different suppliers. Technical interface between subsystems of different suppliers (should support the topic above).

Отформатировано: английский (Великобритания)

ID	Type	Description

Отформатировано: английский (Великобритания)

CHAPTER 3 Constraints

3.1 Basic Rules

3.1.1 It shall be possible, that delivery units from different suppliers can be installed or deployed separately.

That is, it must be avoided, that some program units from different suppliers have to be linked together before installation.

3.1.2 Interfaces shall be defined in a manner, that dependencies between delivery units of different suppliers are as simple as possible.

This supports a lot of topics:

- parallel development
- independent tests (before integration phase)
- simple simulation of interfaces of other suppliers

3.2 Project Specific Rules

3.2.1 The interface between graphical client and Ironman should be as small as possible.

The primary requirement is that the number of parameters shall be small. At second place the number of methods shall be small.

A small interface has several advantages:

- easier to understand for developers using this interface
- better performance
- better maintainability

3.2.2 XML shall be used at interfaces

3.3 Data Model

3.3.1 The entire data model should be structures in “domains”. Each domain is in responsibility of one specialized subsystem/module.

Отформатировано: английский (Великобритания)

In our concrete situation we already have the “TMN domain”. That is TMN data, which are in responsibility of Ironman. Ironman implements these data in MOs and stores this data in Oracle database.

And we have a “graphic domain”. These are graphical information required to display maps and other graphical representations of TMN data. This domain may require temporary data and/or persistent data. This depends on the requirements and concepts.

E.g. it is possible to draw graphics without persistent graphical data, if there is a layout algorithm requiring only the TMN information.

In other cases information about the drawing need to be kept. An example for that is a drawing, which the user creates or edit from a default drawing provided by the system. E.g. the user can edit a drawing to avoid crossing lines or to make the drawing more realistic, because the system provides (as default drawing) a logical but not a physical view.

There may be some graphic information, which can be seen as part of both domains (e.g. geographical location of equipment, which is used to place equipment on a map. But nevertheless the concept should clearly specify which data is in which domain.

Persistent data of the “graphic domain” shall be separated from the persistent data of the “TMN domain”. Direct data access to other domains is not allowed (instead public interfaces have to be used). The domain structure may be more detailed.

The relations between these domains shall be as simple, as possible and shall follow some rules, which need to be defined. E.g. a simple structure is a hierarchical structure: if one domain knows (and refers to) another domain, it is forbidder to have knowledge in the opposite direction.

With respect to the already implemented Ironman system, this rule implies:

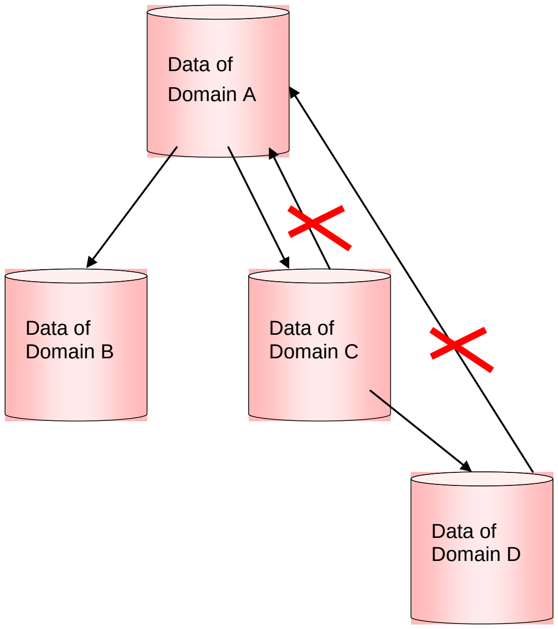
- MOs do not know GOs
- GOs can refer to MOs

Following these constraints we will get a clear structure with several benefits, e.g.:

- Responsibility: data can be changed only by the domain itself or via public interfaces of the domain
- Architecture (modularization): each domain may change its internal data model without influence to implementation to other domains

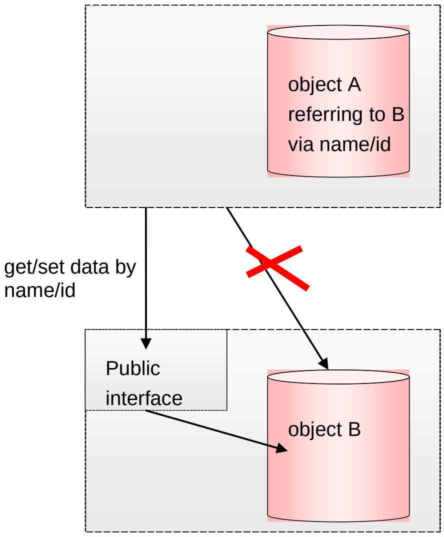
Отформатировано: английский (Великобритания)

Data relations between two domains shall be in one direction only:



Отформатировано: английский (Великобритания)

Data relations between two domains shall be modeled via unique naming/unique keys. Access to data of other domains is allowed via its public interfaces but not via direct database access (e.q. SQL).



3.4 Constraints Out Of Current Architecture and Implementation

3.4.1 Ironman provides BOs as interface objects to TMN information.

As a result graphic clients have to use BOs to access information from MOs.

Remark: The current BO interface will not fulfill all requirements of GOs. This applies to available information as well as for performance issues. Therefore it is strongly recommended to specify an Ironman-BO interface for graphic clients. This interface should be alike the current one (same principles and same basic structure) but adapted to graphical client needs (different content, where required).

Отформатировано: английский (Великобритания)

CHAPTER 4 Analysis of Requirements

4.1 Introduction

This chapter contains a rough analysis of features/requirements derived from the SBS' offer. It is assumed, that this offer is close to the current requirements of PTN discussed with StepLogic.

In a second step the major topics for the SBS/StepLogic interface are derived.

4.2 Common Issues (Standard functionality for all/most of Graphics)

4.2.1 Requirements

- print and hardcopy
- panning (Overview/logical zooming) ??????????????

4.2.2 Interface Topics

- Responsibility for hard copy and print
- Some topic related to Panning???

4.2.3 Data Model Topics

4.3 Passport Manhole

4.3.1 Requirements

- Scheme (default drawing) generated by system containing following information
  - one Manhole
  - Manhole's adjoining Shafts (also called Duck Work Sections)
  - Channel Blocks inside these Shafts
  - Cables (or adjoining Cable Sections??) inside these Channel Blocks
  - Media Joints inside the Manhole
  - Cables (or adjoining Cable Sections??) inside the Manhole connecting Media Joints inside the Manhole??

- edit mode
  - edit vertices (create, move, delete) of lines representing cables (or cable sections) inside the Manhole

Отформатировано: английский (Великобритания)

- add bitmap (no background picture but an additional separate picture of the Manhole)
- save (and quit)
- *… and additional topics not explicitly described in offer:*
- move Media Joints (should be treated alike vertices of cable sections, otherwise you may get strange pictures)
-

Remark: Currently in discussion, whether or not the edit mode is required.

4.3.2 Interface Topics

- ➢ Switch between the default drawing and the edit mode
  - Which part calls/uses with part of the other supplier?
  - How the graphic module takes over the default drawing.

4.3.3 Data Model Topics

- ➢ Scheme (default drawing) generated by system
- ➢ Displayed items (per picture):
  - one Manhole
  - Manhole’s adjoining Shafts (also called Duck Work Sections)
  - Channel Blocks inside these Shafts
  - Cables inside these Channel Blocks
  - Media Joints inside the Manhole
- ➢ Items representing Manholes, Cable Sections and Shafts are one to one related to Ironman Objects
- ➢ Additional pure graphical items: vertices on cable sections

Conclusion:

- ➢ Lot of Iroman data required (objects and its relations)
- ➢ Additional graphical data required only for items inside the Manhole
- ➢ Graphical objects representing Media Joints (inside Manhole) (can) refer one to one via unique id or name to Ironman Objects.
- ➢ Vertices on Cables (or cable Sections) inside the Manhole are pure graphical information not related to Ironman objects. The entire set of lines and vertices for one Cables (or cable Sections) inside the Manhole can refer to one Ironman Object. (to check)

Still open questions:

- ➢ The editing is restricted to item inside the Manhole. Should the graphic store the (other) “surrounding” items and its relations to Ironman objects or should it store the entire surrounding as one picture and add the graphics for the inner part of the Manhole?
- ➢ How could an implementation of both alternatives look like?

Отформатировано: английский (Великобритания)

4.4 Main Cable Scheme

4.4.1 Requirements

- edit mode
    - assign cables, media system, media elements and media joints to main cable
    - edit (create/delete/assign) cables, media system, media elements and media joints to main cable
    - open templates from graphic items in order to edit cables, media system, media elements
    - extra pane for edit attributes of items (cables, …) ?????????
    - save (and quit)

Remark: Currently in discussion, whether or not the edit mode is required.

4.4.2 Interface Topics

- Extra pane for editing a subset of attributes really required or can the complete template be used?
- Update (e.g. via notification mechanism) required in order to have a consistent view on all data, which are displayed in graphics and templates or other windows?
- To do

4.4.3 Data Model Topics

- To do

4.5 Connection Line Cable Scheme

Alike Main Cable Scheme

4.6 Inter-DF-Cable Scheme

Alike Main Cable Scheme

4.7 Passport of Distribution Cable

4.7.1 Requirements

- Generated scheme, no edit mode

Отформатировано: английский (Великобритания)

4.7.2 Interface Topics

- To do (nothing special?)

4.7.3 Data Model Topics

- To do

4.8 Cable Canalization on City Map

4.8.1 Requirements

- edit mode (no generated drawing)
- open template for Manhole in order to create new manholes
- open template for Shaft
- split Shaft, which also uses Manhole template to edit attributes of the inserted Manhole
- move Manhole

4.8.2 Interface Topics

- To do (nothing special?)

4.8.3 Data Model Topics

- TMN data
    - Manhole
    - Shaft
    - Relations between Manholes and Shafts
- Graphical data:
    - position of manhole
    - no further information required for position of Shafts (just straight lines between Manholes, which the Shaft connects)
    - no vertices on Shafts
-

Отформатировано: английский (Великобритания)

4.9 Scheme of DF-Region

4.9.1 Requirements

- xxx

4.9.2 Interface Topics

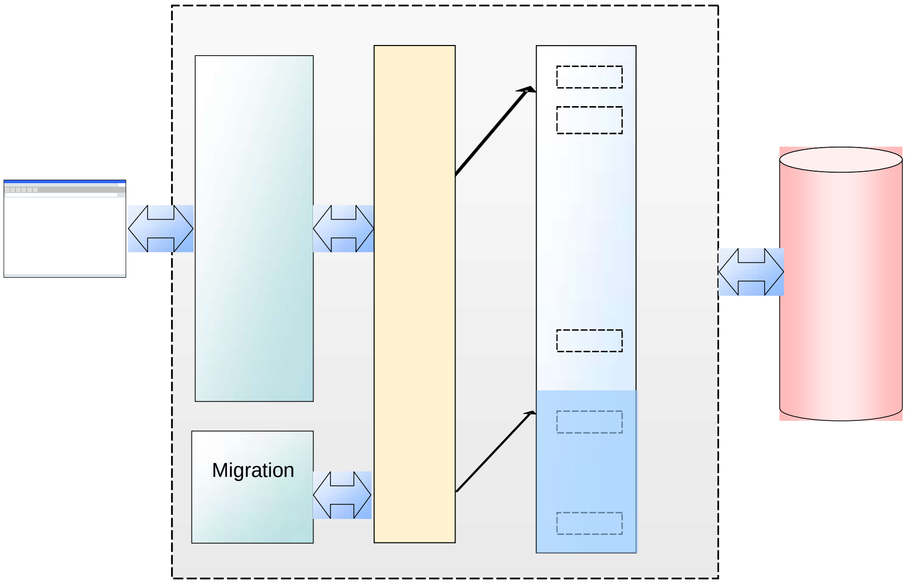
4.9.3 Data Model Topics

Отформатировано: английский (Великобритания)

CHAPTER 5 Solution

5.1 Architecture

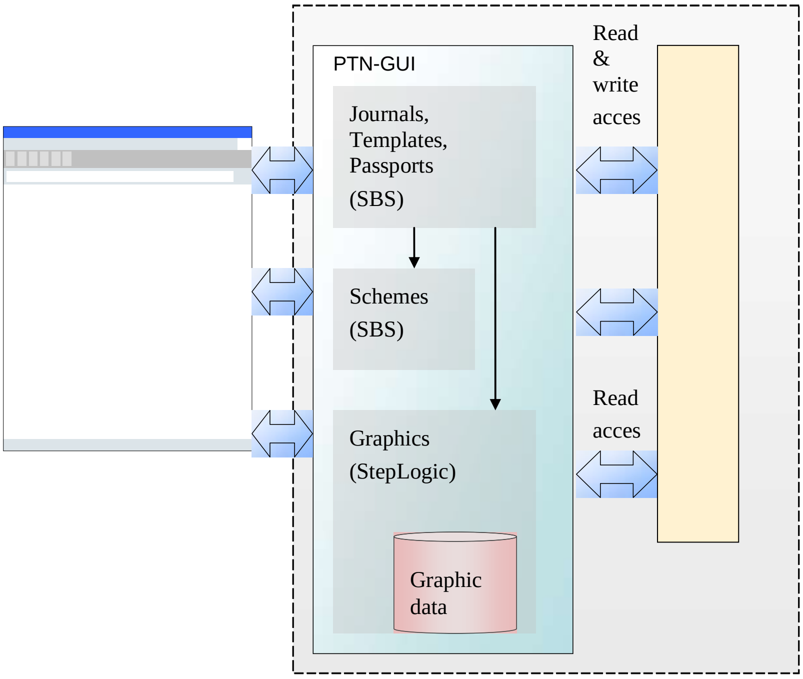
Overall Picture:



Business Objects: special set for graphic clients

GUI Picture:

Отформатировано: английский (Великобритания)



- It is possible to open a Scheme or a Graphic from a Journal.
- Graphic modules can activate other modules. E.g. open some Scheme in order to display details of an item, which is displayed in a map. To do: specify principles of this Interface
- Graphic modules can get information about TMN objects via BO interface; this access should be limited to basic information like existence of data, naming, containment or usage relations.
- Other persistent graphic data shall be stored separately (in an own data model without any direct relation to Ironman DB. Relations between graphic objects and Ironman objects shall be done via naming (or Ids).

5.2 Data Model

Consider different types/pattern of data sources and data relations:

Отформатировано: английский (Великобритания)

For graphs:

- A node in the map/scheme is directly related to one IR object
- An edges in the map/scheme is directly related to one IR object
- An edges in the map/scheme is one relation between two IR objects
- A set of edges in the map/scheme is a set of relations between several IR objects
- A node in the map/scheme is not related to any IR object
- An edge in the map/scheme is not related to any IR object
- Positions of nodes are derived from attributes of IR objects
- Positions of nodes are not derived from attributes of IR objects
- Icon type and size (e.g. diameter or height) of nodes are derived from attributes of IR objects
- Icon type and size of nodes are not derived from attributes of IR objects
- Line attributes (e.g. shape, length) of edges are derived from attributes of IR objects
- Line attributes of edges are not derived from attributes of IR objects

For boxes (required for some graphics?):

- A box is directly related to one IR object
- Containment of two box is directly related to a (containment) relation between two IR objects
- A box is not related to any IR object
- Containment of two box is not related to any (containment) relation between two IR objects
- Box attributes (e.g. color, size) are derived from attributes of IR objects
- Box attributes are not derived from attributes of IR objects

5.3 Interfaces

Отформатировано: английский (Великобритания)

CHAPTER 6 Open Points

6.1 Architecture

- Inter-working of Graphics with Journals, Templates, Passports and Schemes

  Graphic modules implemented by StepLogic shall use SBS-modules to display and change TMN data.

  I do not know whether or not we need this inter-working. But if we need it, we need to have a concept, how Journals, Templates, Passports and Schemes can be opened by the StepLogic graphics.

- Security/Session Context

  How to come to a single password check, a valid login for all BO requests and a common local session context?

6.2 Data Model

6.3 Interfaces

6.4 Requirements on Performance and Throughput

For each type of scheme/map the expected amount of data need to be specified. Either this is a user defined scope (the user has to respect the recommended amount in order to get and keep reasonable performance) or there is a rule implemented by the system. In the first case before design phase this amount of data has to be specified (just as a limit). In the latter case this rule has to be defined and the resulting amount of data has to be calculated and documented. Anyway these numbers are a prerequisite and design decisions will depend on these limits.

Отформатировано: английский (Великобритания)

Отформатировано: английский (Великобритания)