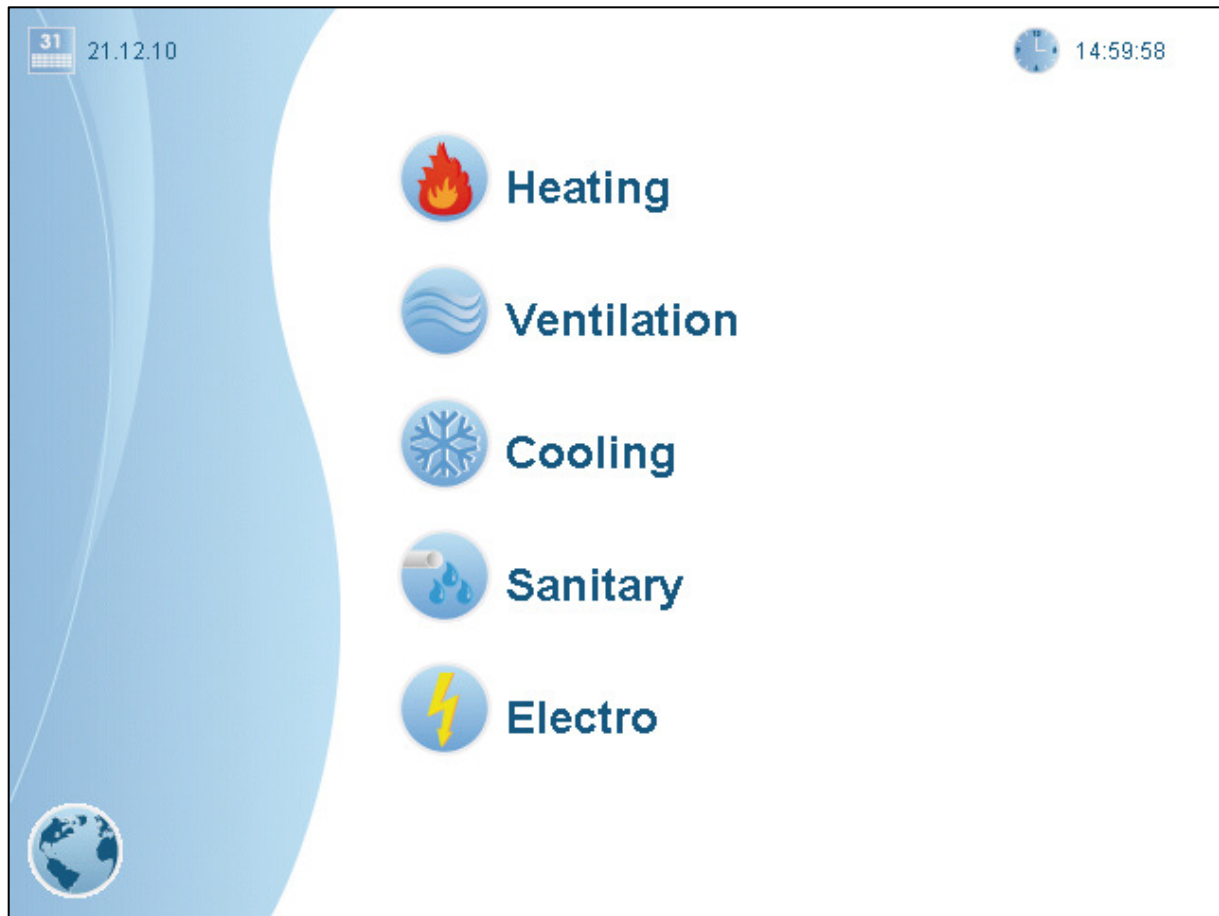


Getting started with the HVAC Demo Application



Project history

Date	Author	Modification
21.12.2010	TCS / gm	Creation of documentation (version E1) and project for PG5 2.0.150
05.04.2011	TCS / gm	Final version for publication (E5)

Content

1 Introduction.....	3
2 Required hardware and software.....	3
3 Connecting to the demo application.....	3
4 Project structure	4
4.1 The fupla structure	5
4.1.1 Common part.....	6
4.1.2 Function specific part	6
4.1.3 Function block structure.	7
4.2 Symbol structure	12
4.3 Fupla templates.....	13
5 WEB application	15
5.1 Icon library.....	15
5.2 WEB-application structure	16
5.3 Usage of WEB-macros.....	17
5.3.1 Complete system macros.....	18
5.3.2 Regulation, control, adjustment macros.....	20
5.3.3 Trending and alarming macro.....	21
6 References	23

1 Introduction

This document serves as a simple introduction to the use of HVAC Demo Application.

This document in application with the associated PG5 2.0 project can be used to understand how easy to work and visualise with Saia fupla programs and WEB-Editor in case of using well structured fupla programs, symbol hierarchy and the related Web macro-templates.

The information contained in this document is just a snapshot of the most important things you should know to understand the easy and efficient workflow with Saia-software. For more information, please refer to the corresponding documents (see section "References").

2 Required hardware and software

The only thing you need to check the application running on a PCD in Murten is a running PC with [Java](#), internet browser and internet connection.

In case of you would like to check the program-code you will need an installed [PG5 2.0.150 or later with S-Web Editor 5.14.30 or Higher](#) to be able to open the application project.

It is also possible to check the project on your own PCD. For that, please check the following document which describes how to work with PG5 and WEB-editor. (GettingStarted_MBPanel_VGA_E2.doc)

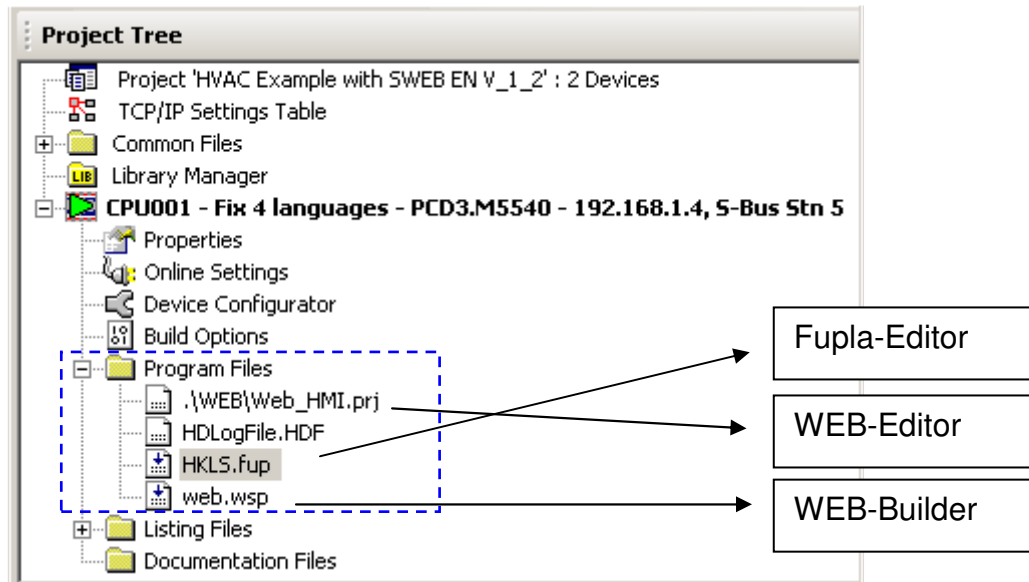
3 Connecting to the demo application

To access the HVAC demo application you should click on the following link:
http://s-web-demo.sbc-support.ch/Start_en.html

4 Project structure

To create an application you checked in the link before you will use the programming tool of Saia named PG5, the current version is 2.0.150.

The project tree of the demo application is shown on the picture below.



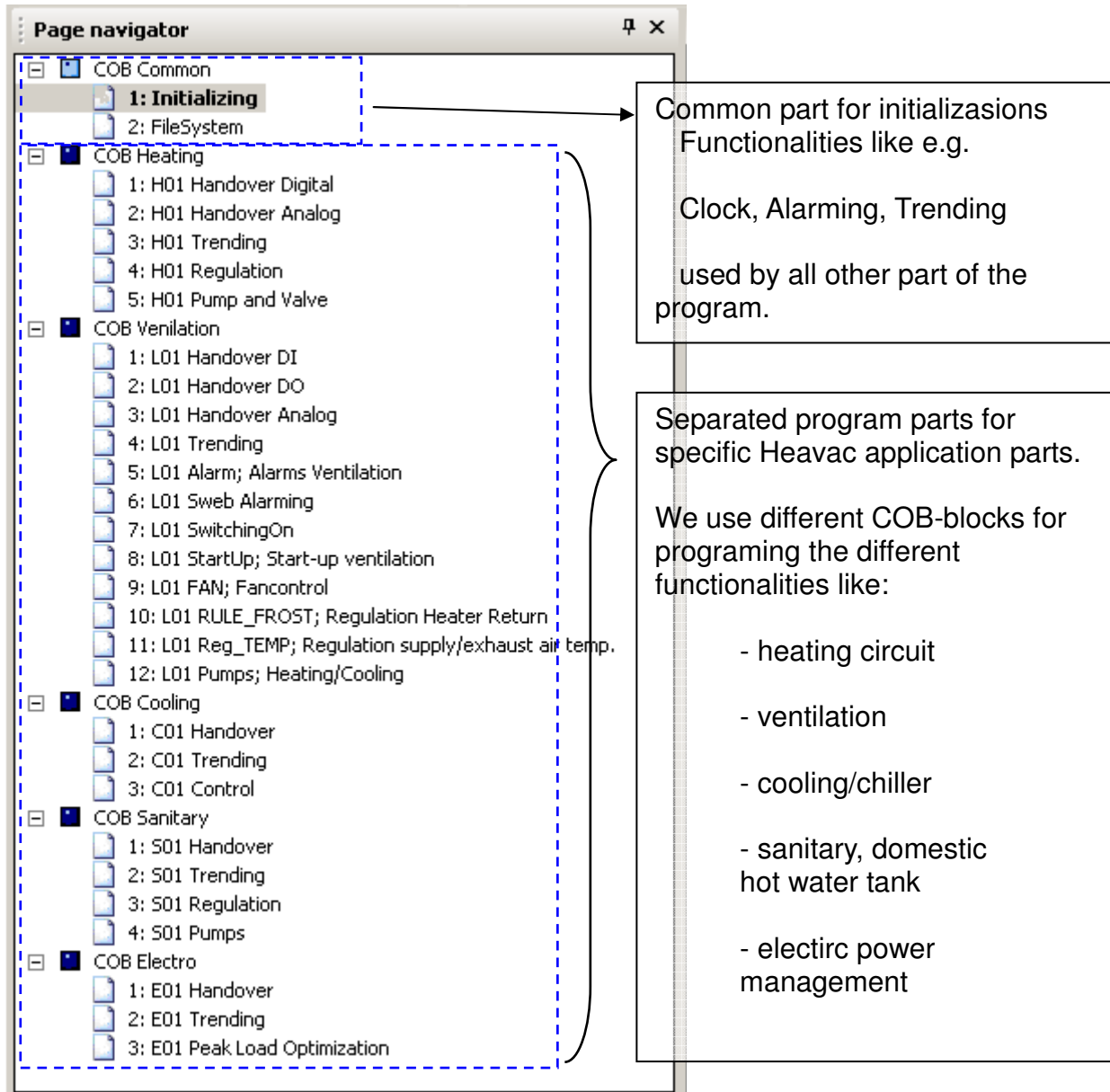
In this document we concentrate only the **Program Files** folder of the project.

This contains the files for defining/describing

- the program algorithm to manage the HVAC tasks controls ("HKLS.fup"), editable with Fupla-Editor
- the visualisation ("Web_HMI.prj"), editable with the Web-Editor
- the Web-server functionality ("web.wsp"), editable with WEB-Builder.

4.1 The fupla structure

The program structure of our fupla file is shown on the picture below.



To be able to easily manage our application it makes sense to divide it into functional parts. These parts are organised in separated continuously running program-blocks (COB) and are responsible for different Heavac tasks.

4.1.1 Common part

The first program-block we usually use for common things, so contains the functionalities we are going to use in all other program parts.

These functionalities are:

- Initialising the heavac functionalities (reset, clock, Summer-Winter change, performance control of CPU, detection of general acknowledgment)
- Initialising and configuring the Alarming,
- Initialising and configuring the File system for trending.

4.1.2 Function specific part

The following program-blocks are containing the program-logic of the different heavac functionalities.

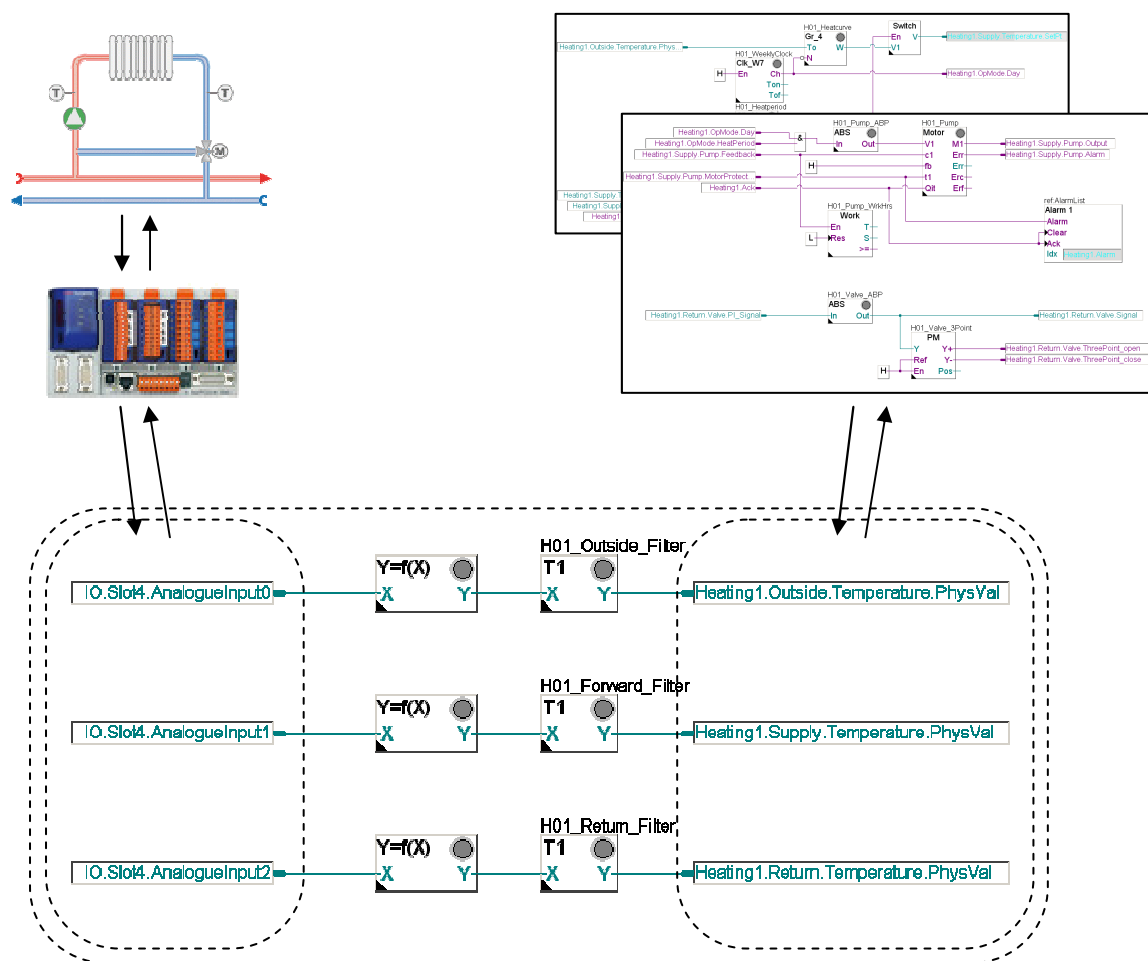
- **Heating**
This program block contains a simple program algorithm to manage a single heating circuit, regulate the temperature and control the corresponding valves, pumps.
- **Ventilation**
This program block contains a simple program algorithm to manage a ventilation system with cooling and heating heatexchanger and heatrecovery unit, regulate the room temperature via supply air temperature and control all corresponding valves, pumps and fans.
- **Cooling**
This program block contains a simple program algorithm to manage a chiller responsible to providing cold energy to other systems.
- **Sanitary**
This program block contains a simple program algorithm to manage the sanitary system to provide domestic hot water for household. It regulates the boiler temperature via supply heating water temperature, controls the corresponding valve, pumps.
- **Electric**
This program block contains a simple program algorithm to manage the power via switching off some equipments in case of power-peaks.

4.1.3 Function block structure.

As you can recognise on the previous picture all functional block consist of more pages.

- Handover**

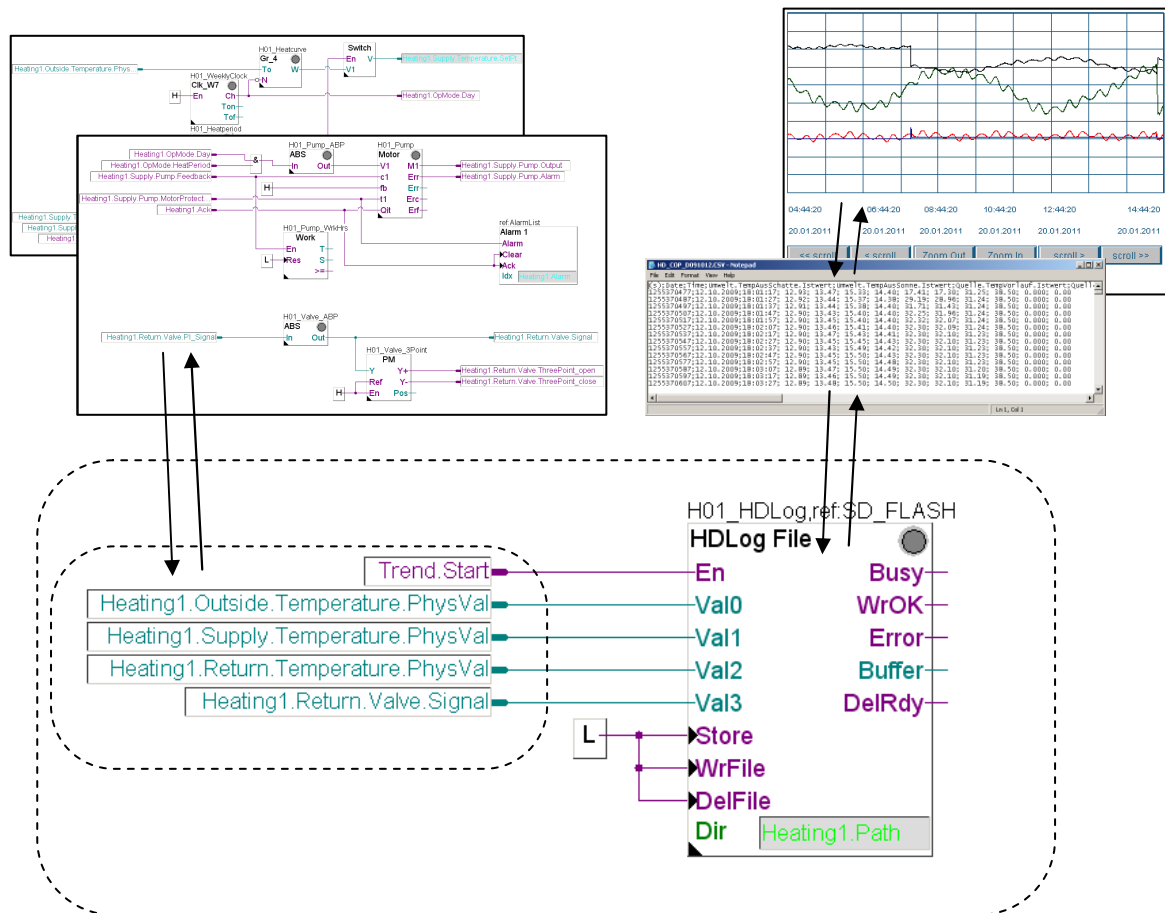
The first few pages are always handover pages for linking the physical I/Os. Here you can connect the PCD-resources containing the analogue and digital input and output values to the logic responsible to manage one of the before mentioned functionalities.



- **Trending**

The pages after the handover pages are responsible for collecting all necessary data we would like to log and be able to show in trends.

so it gets all of the current data from the program-logic and saves to a CSV-file to be used for Historical trending.

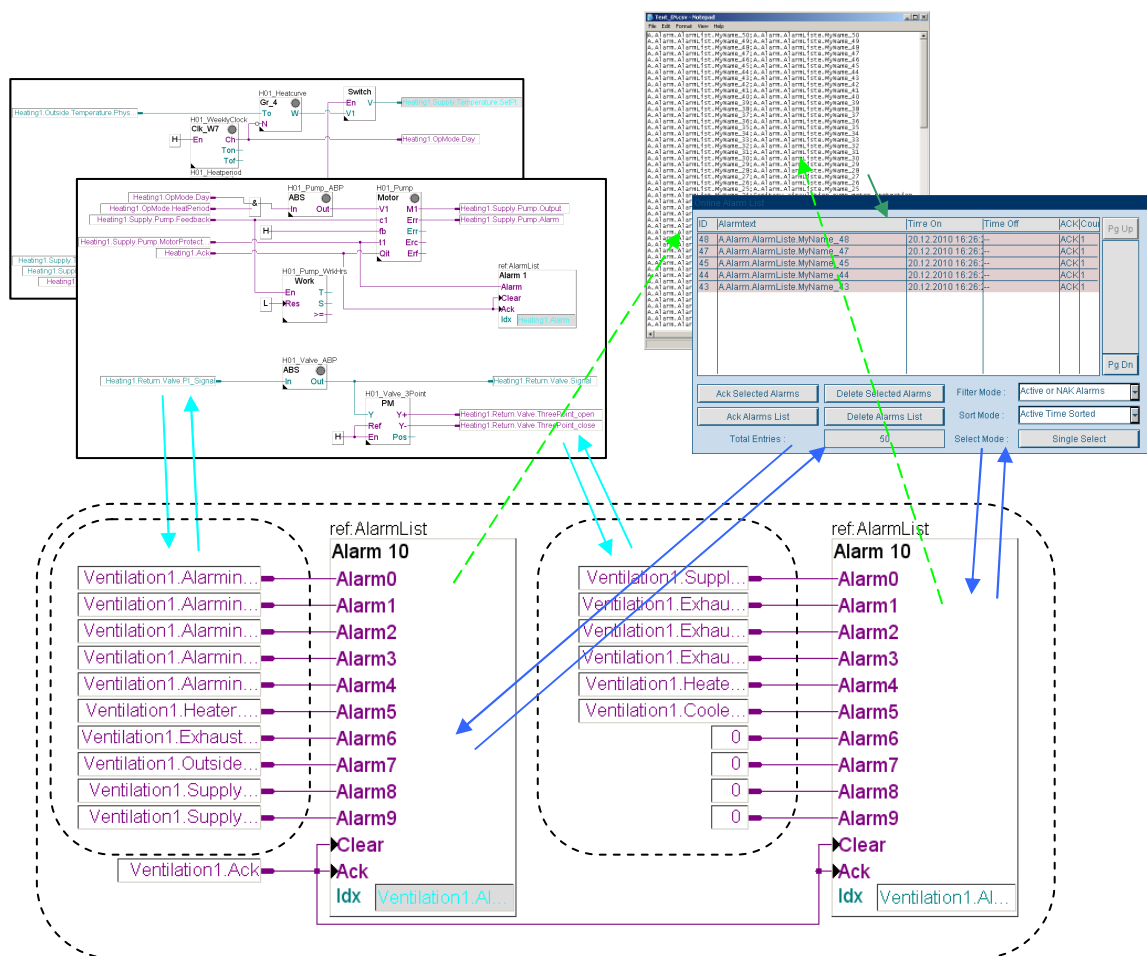


Alarming

In the most complex function block you can find some additional pages are responsible for collecting all necessary data we would like to use for alarming.

so it saves the alarming text information to a CSV file during the build, and in runtime gets all of the current alarm-status data from the program-logic.

with help of the runtime collected alarm status flags and the corresponding alarming texts located in the CSV-file was created during the build we can easily visualise the alarms.



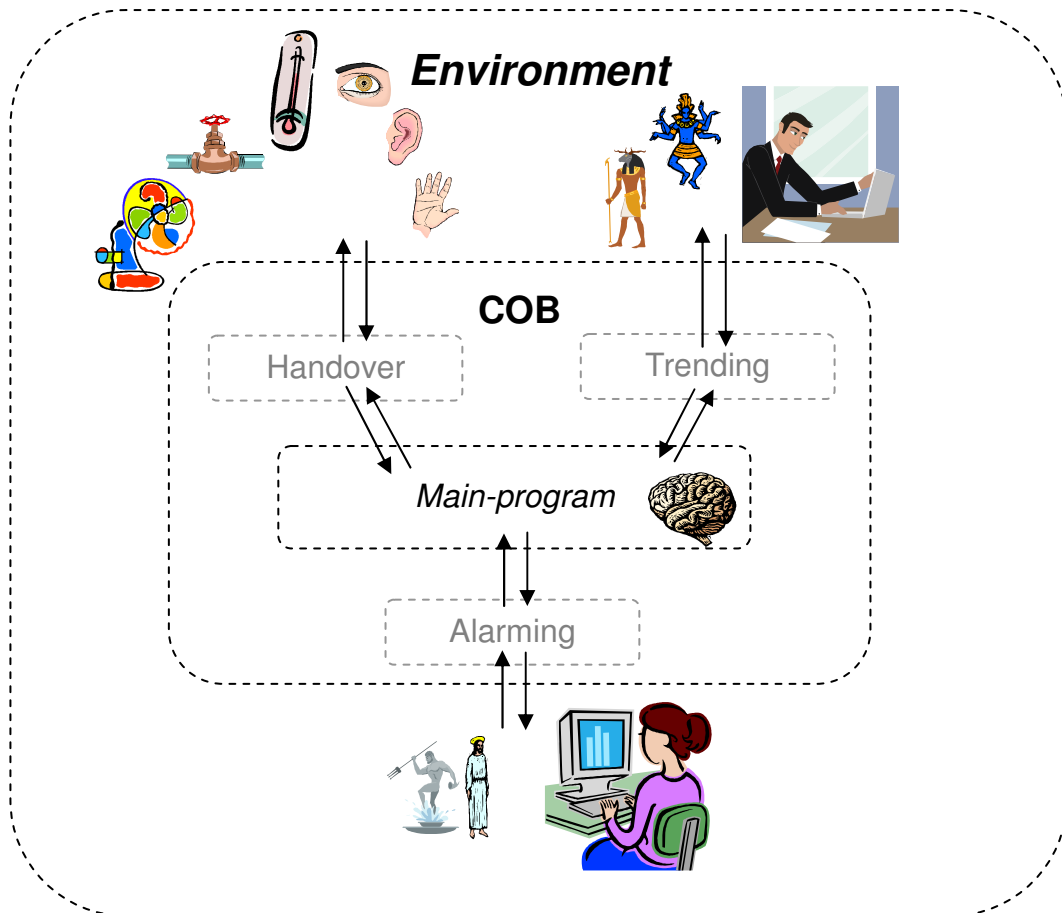
- **Regulation/Control**

As you surely recognised all of the functionalities we mentioned in the previous points are only necessary extensions to help some kind of main program part or “brain” (we called program-logic or algorithm) to be able to communicate with its environment.

So the **Handover** program part is responsible for linking the senses and the limbs (sensor and actuators) to the “brain”, hereby providing connection to a part of the “universe” was assigned to manage it.

The **Trending** program part helps the “brain” to prove the “god(s)” that it was perfectly able to fulfil the task was delivered, or in case of abnormalities it is used for helping “god(s)” to find out the reasons and fix the problems.

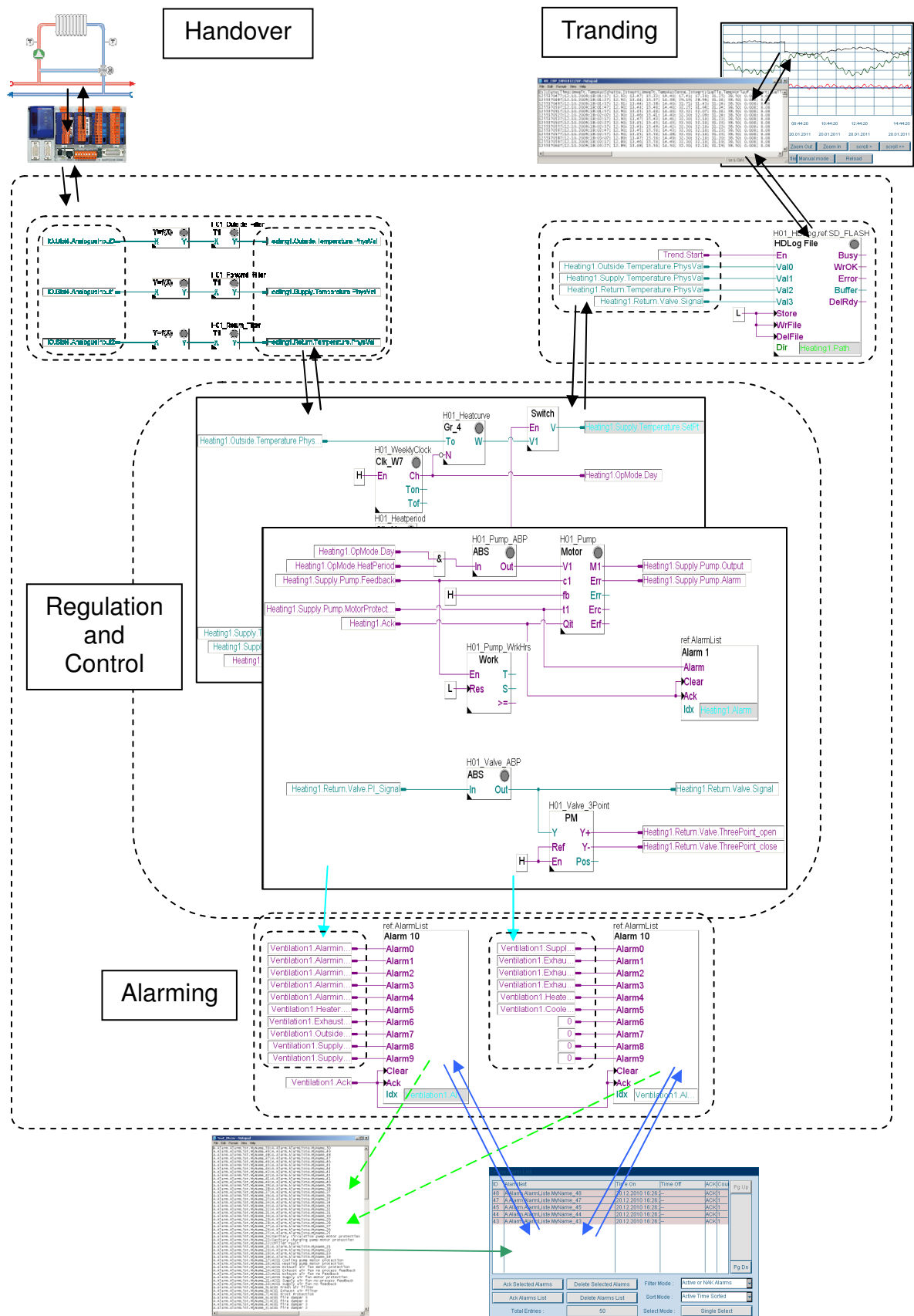
The **Alarming** program part helps the “brain” to provide more detailed and higher level information for “god(s)” about his status, about the disorders, errors, faults can hinder the accomplishment of its main task.



The main program itself contains the program code for regulating the necessary parameters and control all of the equipments are used to fulfil the functionality delivered to it.

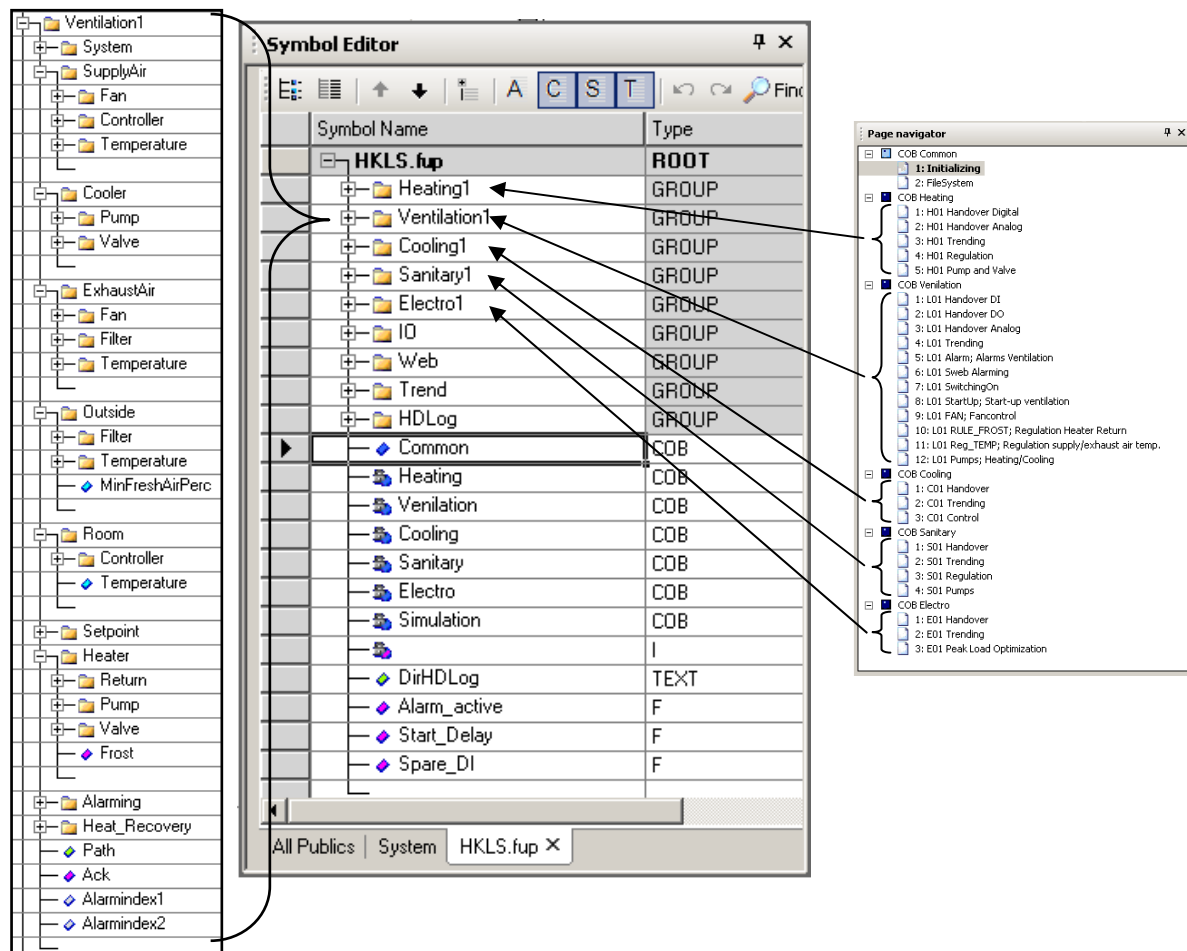
So e.g. calculates the current setpoint for the temperature control, regulates the temperature with a 3-way analogue valve, starts/stops the pump(s), fan(s), open/close the dampers...

The next picture summarises the internal and external connections between the different parts of one heavac functionality and it's environment.



4.2 Symbol structure

The structure of our symbols used by fupla is shown on the picture below.



The first things you can recognise that we organised the symbols to the same group-structure as the fupla so all symbols used by e.g. the Heating COB are collected into a symbol group named "Heating1".

We use the index "1" because there can be more then one heating circuit in the application and as you see the pages of the COB contains the program of this only one heating circuit (the H01-prefix identifies it).

To other tasks managed by the other COBs belong their own symbol-group as well, and because we have only one task pro COB in this demo application we use in all symbol-group names and COB-page names the index "1".

If you have a larger application with plenty of heating circuits, ventilation systems, chillers then you will use additional indexes to differentiate them from each other.

Afterwards you can see that the main groups containing the symbols for the different HVAC tasks consist of additional subgroups.

The picture above contains the collapsed main group of Ventillation1.

You can see that the subsystems have their own subgroup (cooler, heater, supply air, exhaust air).

4.3 Fupla templates

The previous two chapters described you how we organised our fupla program and the corresponding symbols in well structured hierarchy to help our work during the program-coding and make our life easier during the maintenance.

But beside these advantages we can use it for a more powerful work in case of we have bigger application or we would like to use it later on in an other application.

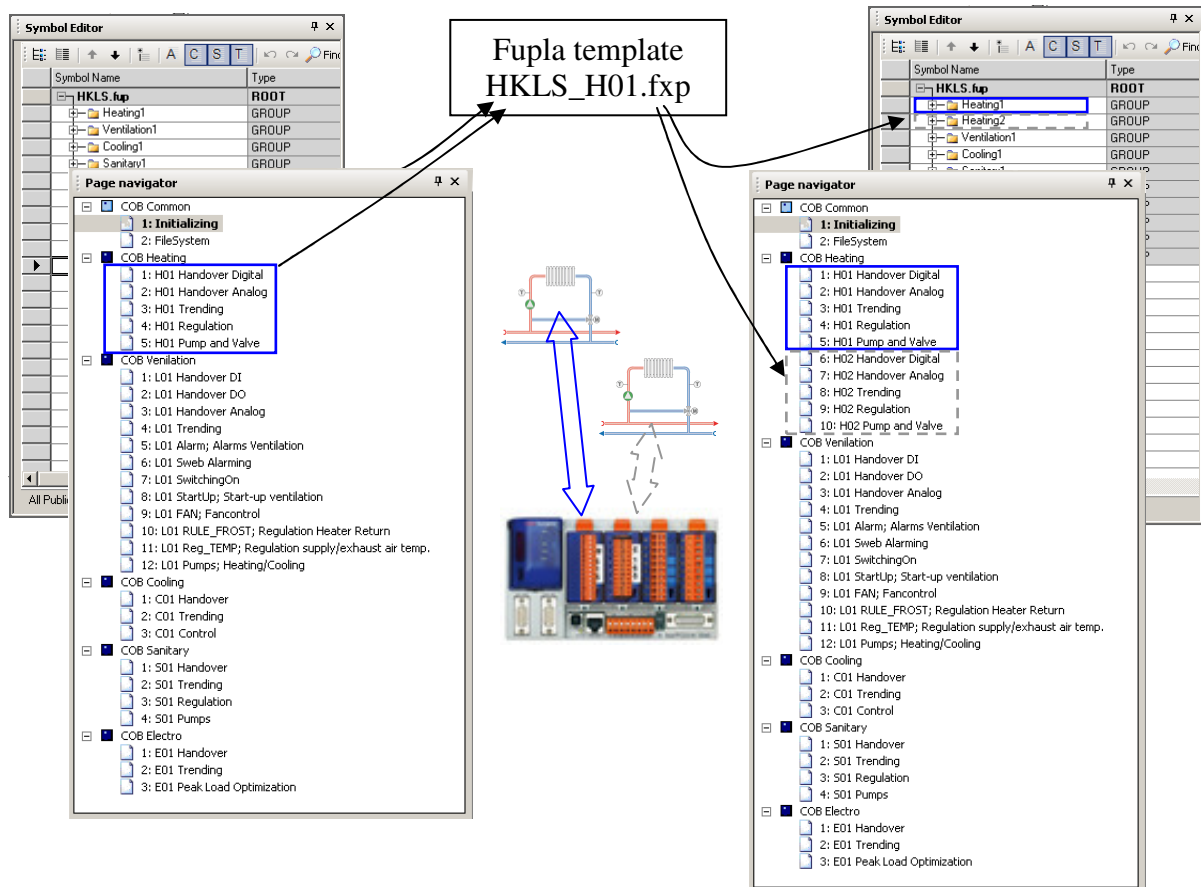
If we would like frequently use the same functionality then we can create templates from those.

To create a template from a program part you just have to export the fupla pages you would like to use frequently.

Afterwards you can import it in many instances with automatically re-indexed fupla page names, fbox-names, symbol group names, pcd-resource addresses.

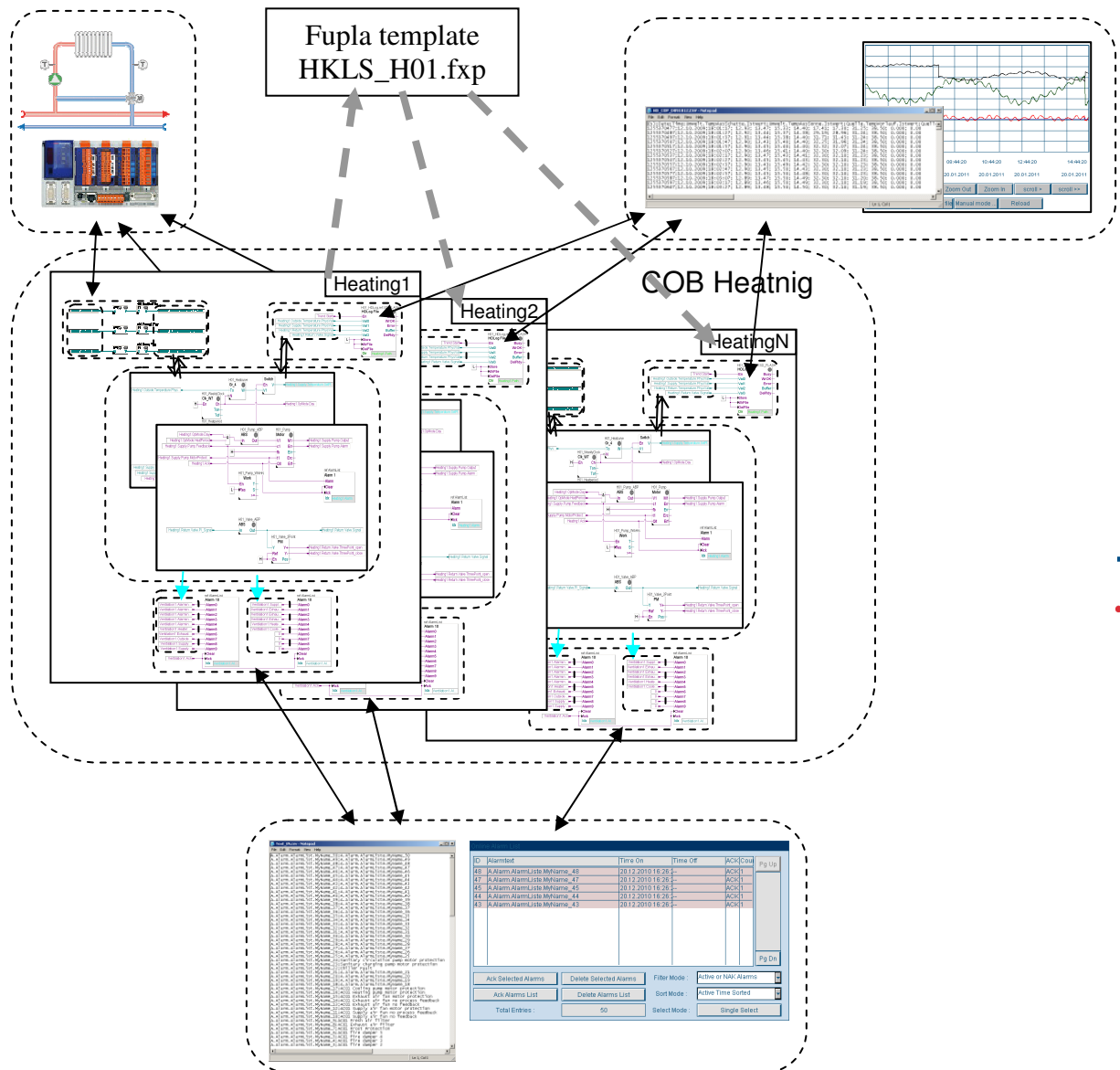
These imported pages contain all of the functionalities we mentioned before (handover, trending, alarming, and regulation/control).

The only thing you should do is to actualise the physical I/O links on the handover page.



With this template mechanism you can very powerfully create plenty of instances of the same functionality in a few minutes with a few mouse clicks.

The newly inserted fupla pages containing the imported template instances are containing the complete functionality of the original fupla-pages served as source of the template, and have the same symbol structures like was created for the template.



5 WEB application

In this session we are going to manifest you our workflow we used to create the WEB-part of the demo application.

5.1 Icon library

Saia created, developed plenty of useful nice pictures and icons to commonly use in different visualisation applications.

Saia used these pictures for the new design in WEB-editor and VisiPlus-GraphicalEditor as well. The pictures are available in different sizes and formats and are going to be accessible from the [support page](#).

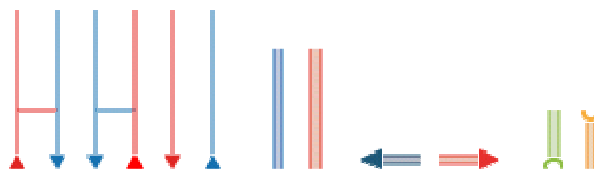
There are pictures for special navigation/menu buttons:



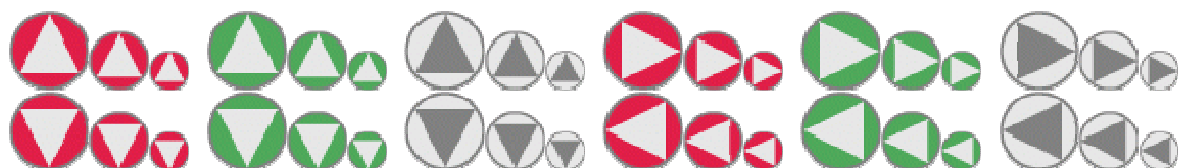
for heavac equipments like sensors, pumps, fans, valves, heating/cooling heat-exchangers:



for different pipes:



Beside the different sizes for different application resolution and formats for different application types the pictures are available in different colours and directions to be able to build all possible heavac application types.



5.2 WEB-application structure

We know already what kind of pictures we used for our demo-application.

We know what kind of heavac functionalities we programmed in our fupla-program; we called them Heating, Ventilation, Cooling, Sanitary and Electric.

We know that all of these has trending, and some of alarms.

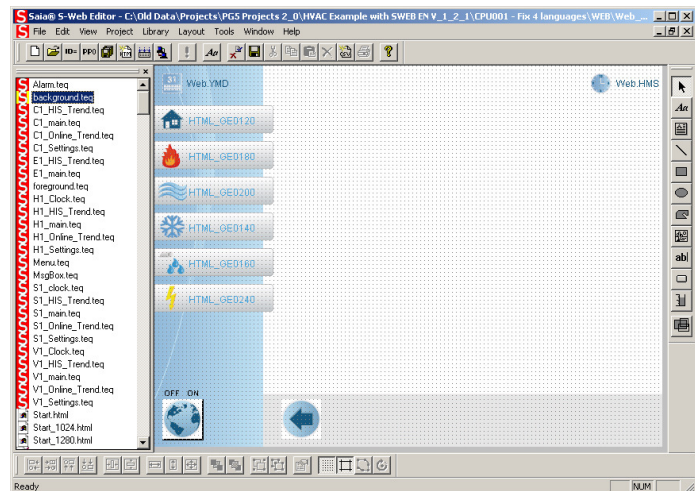
We know that all of these has a regulation and control part.

Finally we know that we would like to show the application for customers of different nations.

So we decided to create a background picture what we used for the navigation and visualisation elements we use on all of our web-pages.

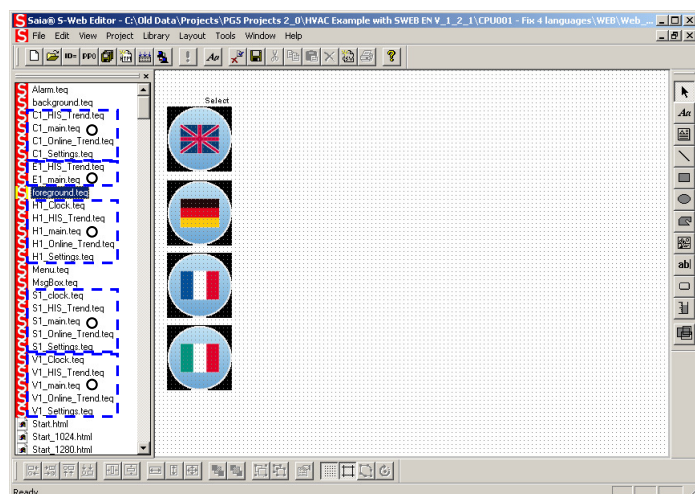
These are

- the background picture with blue and gray areas for the navigation/menu buttons,
- the date and hour values,
- the language button to popup some kind of language selection dialog,
- the main navigation buttons to change between heavac application/task types,
- the “back” button to jump to previous page we saw.



We put the language selector popup dialog to the foreground page we are linked to all other pages as foreground. So we solved the language selection on this single page.

All other schemata pages you could already recognise in the project files list on the left side of the WEB-editor are using this two common pages for drawing the background below and the foreground on top of their own content.



5.3 Usage of WEB-macros

In the 4th chapters we showed you how we organised our fupla program- and symbol-structure.

We explained that in this way you can easier work during development, and check the program during maintenance.

We emphasised that if you work this way than you can dramatically decrease the programming time in case of plenty of similar tasks in the same project because you can use your original program-code as template for the other parts of the program or you can use one of your former coded templates via a simple import procedure.

But this workflow has an additional advantage.

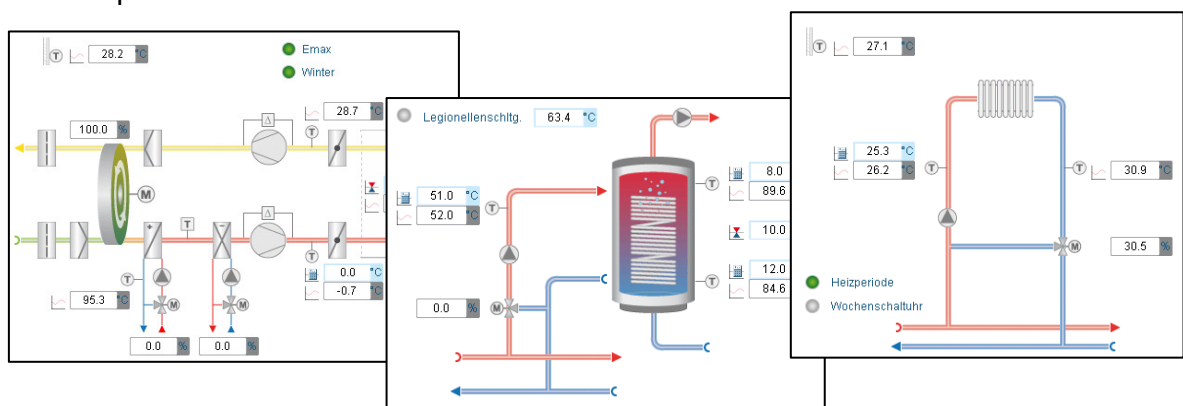
Visualise them with our WEB-technology is like a child's play.

We have plenty of useful WEB-macros to make your visualisation procedure easier.

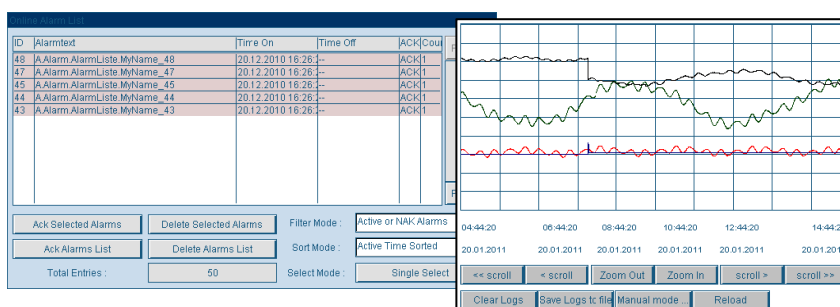
There are macros for single heavac equipments like pumps, valves, dampers, fans:



for complete heavac tasks:



and for special functions like alarming and trending:

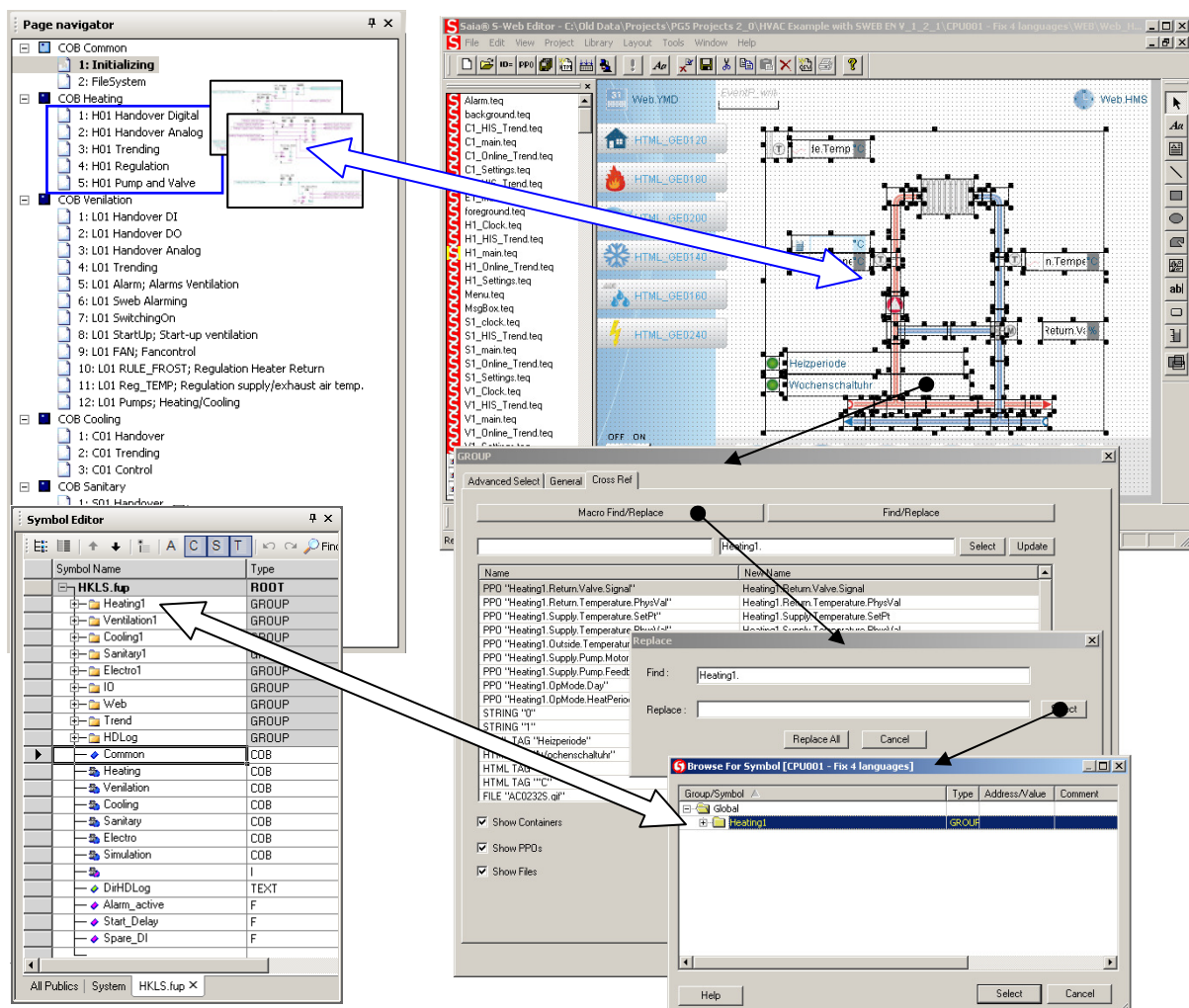


5.3.1 Complete system macros

In our demo application we used these original macros from the WEB-editor macro folder and adapted to our HVAC program parts.

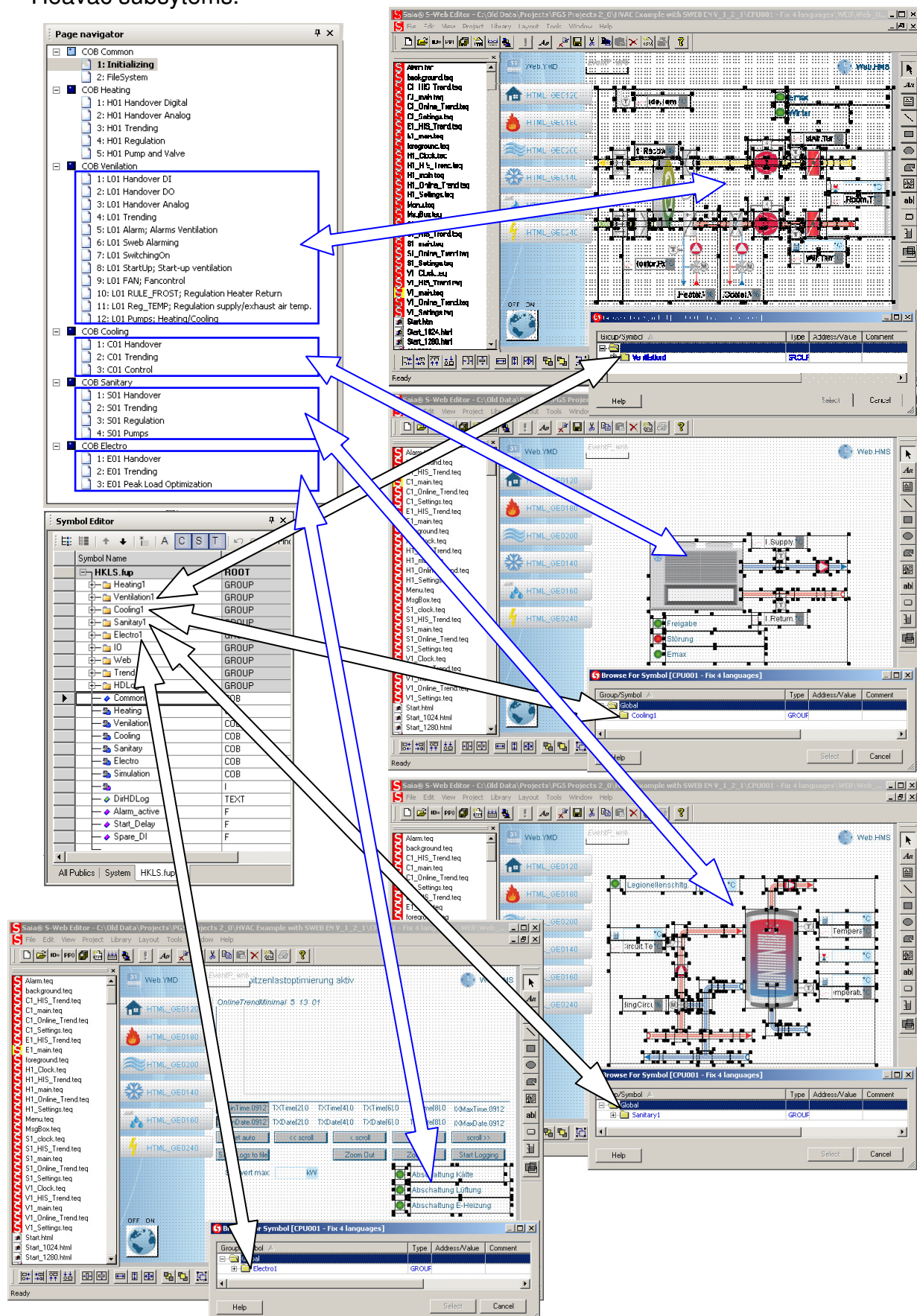
So we connected our heating circuit symbols to our heating circuit WEB-macro. To assign symbols used by PG5 to WEB-macros the WEB-editor searches the symbol-editor and tries to find symbols in it having the same structure as the symbols in the used WEB-macro.

So in our case because we adapted an original WEB-macro and changed the symbol references in it according to our new fupla-template symbol structure the WEB-editor automatically recognises our only heating circuit and offers to use for the macro-instance we inserted to our page in WEB-editor.



During insertion procedure of a WEB-macro the WEB-editor creates all necessary connections to the selected PCD-recourses, inserts all pictures to the WEB-project to be able to visualise it.

We used the same procedure for all other parts of the demo application. As you can see the WEB-editor recognised the symbol groups created and used during programming fupla, so was really easy to create the main pictures of the Heavac subsystems.



5.3.2 Regulation, control, adjustment macros

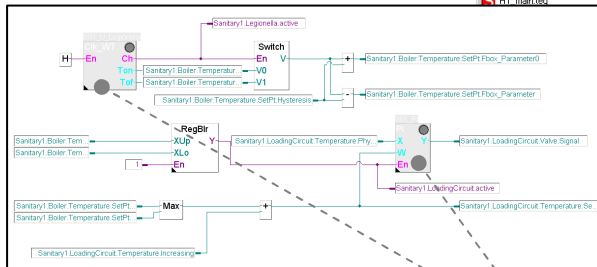
But as we know beside the nice schemata pictures of our functionalities we would like to adjust the regulations and some timers used for control, manipulate the valves, pumps in case of demand.

Fortunately we have all of the WEB-macros are representing this functionalities, and because of course they use the same mechanism to assign symbols from PG5 the WEB-editor will automatically recognise these symbols.

The next picture shows you tree WEB-macros inserted to the clock and settings subpages of the Sanitary WEB-page.

The first window is the Sanitary schemata...

The second is the fupla-page containing the "Weekly clock" and "PI-controller" fboxes.

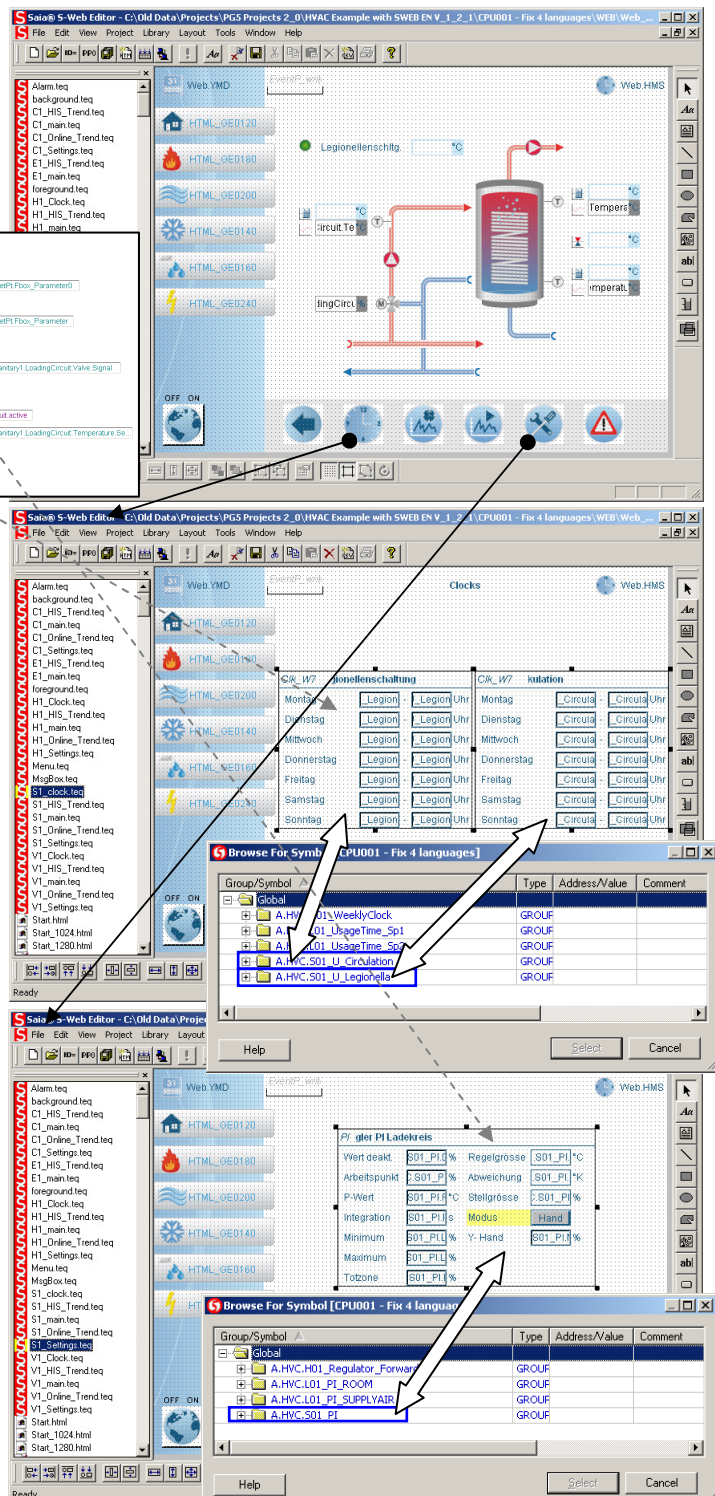


On the third, what will be accessible via the clock-icon on the bottom of the web-application are two „weekly clocks” for adjusting legionella and circulation pump activities.

Because there are more symbol groups representing the data of a „weekly clock” for other parts of the application we selected the right-ones from the list.

On the fourth, what will be accessible via the screwdriver-icon on the bottom of the web-application is one „PI-controller” for regulating the temperature of the supply circuit.

Because there are more symbol groups representing the data of a „PI-controller” for other parts of the application we selected the right-one from the list.



5.3.3 Trending and alarming macro

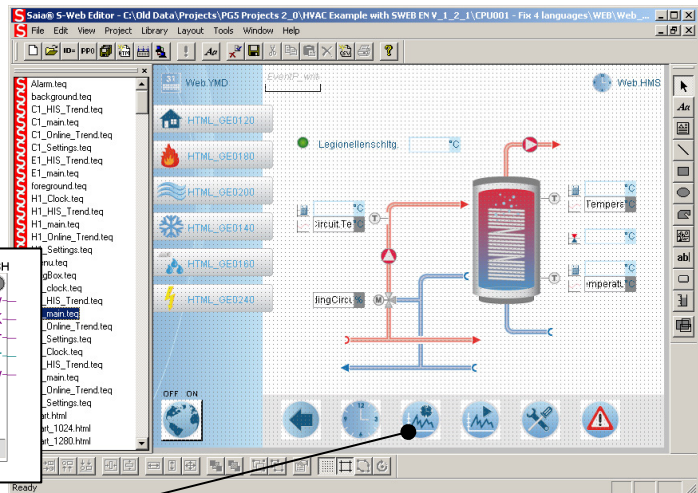
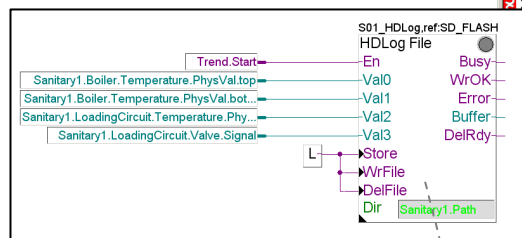
Last but not least we have to visualise the diagrams and alarms managed by the corresponding parts of the fupla-templates in our web-demo project.

Of course WEB-editor has the right WEB-macros to be able to easily solve this.

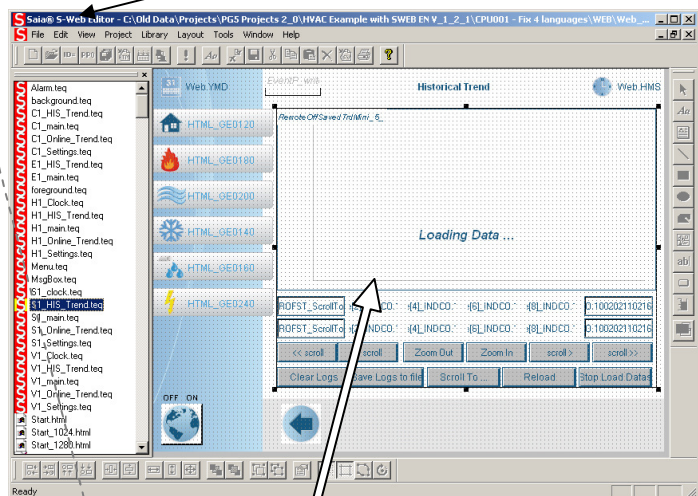
The next picture shows you the web macro inserted to the Historical trending subpage of the Sanitary WEB-page.

The first window is the Sanitary schemata...

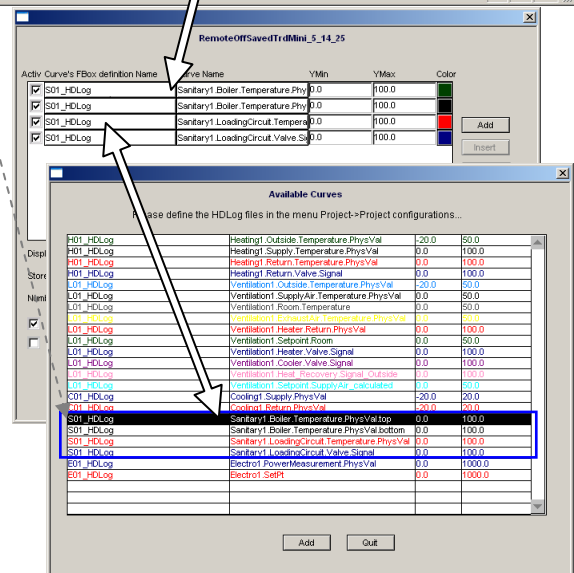
The second is the fupla-page containing the related HDlog fbox.



On the third, what will be accessible via the first trend-icon on the bottom of the web-application is a "Historical-trend" for visualizing the values were linked to the HDlog-fbox in the sanitary fupla-template was showed in window two.



The fourth window is the adjust window of the "Historical-trend" macro already filled with the right parameters selected from the list of all available curves shown on the fourths window.



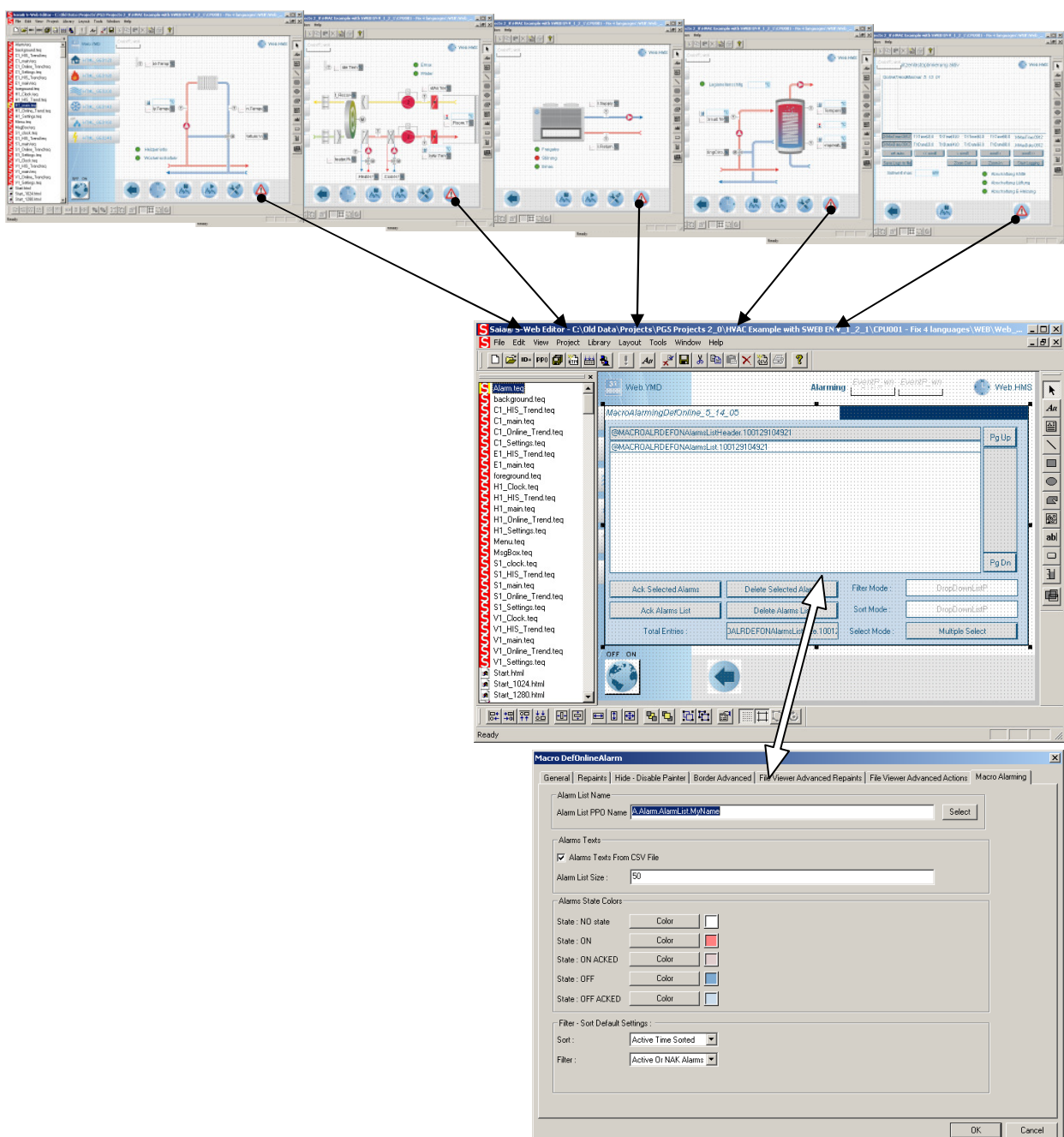
The "Available curves" list was created automatically during the build and contains all of the variables was linked to any of the HDlog-fboxes in our demo application.

So here we selected the symbols from the "Sanitary1" symbol group.

The alarming page is common for all application parts because the alarming macro connects to the “alarmlist” fbox in the common part of the fupla program..

On the picture below you can see that all schemata pictures are referencing to the same Alarm-page, what is accessible via the Alarm-icon on the bottom of the web-application.

The Alarm-page contains the Alarming WEB-macro what is able to communicate with the “Alarmlist” fbox, responsible to visualize the alarm statuses and manage the user actions regarding to alarms (Acknowledge and/or delete alarms or the complete current alarm list).



6 References

Topic	Document	No.
S-Web system	Flyer Saia® Web Panels with Micro-Browser Technology	26/435 E7
MB panel configuration	Manual PCD7.D4xx	26/851 E2
S-Web-Editor	Technical Information Saia® S-Web-Editor	26/453 E2
WebConnect	Manual Saia®.Net & Web-Connect	26/800 E3
Macros	Online Help – Saia® S-Web-Editor	-
Diverse	Saia® FAQ Manager www.sbc-support.ch/faq	-
	GettingStarted_MBPanel_VGA	E2