sCO2-4-NPP: Innovative sCO2-Based Heat Removal Technology for an Increased Level of Safety of Nuclear Power Plants

Deliverable 6.2
sCO2 System integrated
in PWR-simulator

Work programme topic addressed: NFRP-2018-10: Encouraging innovation in nuclear safety for the benefit of European citizen
Type of action: Innovation action

Grant Agreement number:	847606
Start date of project: 1 September 2019	Duration: 36 months
Lead beneficiary of this deliverable:	KSG
Due date of deliverable: 31/01/2022	Actual submission date: 28/02/2022
	Version #: R1.0

Type		
R	Document, report excluding the periodic and final reports	X
DEM	Demonstrator, pilot, prototype, plan designs	
DEC	Websites, patents filing, press & media actions, videos, etc.	
OTHER	Software, technical diagram, etc.	
Dissemination level		
PU	PUBLIC, fully open, e.g. web	X
CO	CONFIDENTIAL, restricted under conditions set out in Model Grant Agreement	

Revision History

Release	Date	Reason for Change	Author	Distribution
R0.1	31.01.2022	Internal Review	P. Lasch	Consortium
R1.0	28.02.2022	Final Delivery	P. Lasch	Public

Deliverable Contributors

Authors

Partner	Name
KSG	Peter Lasch

Contributors

Partner	Name
KSG	Anne-Maria Bleckmann
KSG	Peter Sänger
KSG	Dr. Frank Boes
GfS	Frieder Hecker

Internal Reviewers

Partner	Name
EDF	Albannie Cagnac
USTUTT	Jörg Starflinger

Table of contents

List of Tables

List of Figures

1 List of Acronyms

Abbreviation / Acronym	Description / meaning
API	Application Programming Interface
CHX	Compact Heat Exchanger
CP	Compressor
CVR	Centrum výzkumu Řež
DUHS	Divers Ultimate Heat Sink
FMI	Functional Mock-up Interface
FMU	Functional Mock-up Unit
GfS	Gesellschaft für Simulatorschulung mbH
IC	Initial Condition
KSG	Kraftwerks-Simulator Gesellschaft mbH
NPP	Nuclear Power Plant
sCO2	Supercritical Carbon Dioxide
TB	Turbine
TCP/IP	Transmission Control Protocol/Internet Protocol
VLAN	Virtual Local Area Network
VM	Virtual Machine
WP	Work Package

2 Executive Summary

The goal of Work Package 6 in the sCO2-4-NPP project is to demonstrate the behaviour of the sCO2-Loop in a real power plant. Therefore, a training simulator was chosen as the best way to display the impact of the sCO2-Loop on realistic transients. To achieve this the sCO2-Model developed in WP5 will be connected to a full-scope training simulator.

Deliverable 6.2 aims to describe how this fast-running model of the sCO2-Loop was integrated into a full scope simulator. We will describe all aspects of the code coupling and will explain how the two simulation codes interact with each other. After reading this deliverable the reader will have a clear understanding how the sCO2-Model was connected to the full-scope simulator.

As the interface definition is a well-known process and was already described in deliverable 6.1, the main results will be the description of the interface architecture and the definition of all data points that need to be exchanged for thermodynamic coupling and for display purposes.

We will also focus on the hardware setup needed to execute the sCO2-Model and demonstrate the real time capability by running first transient.

When the integration into the simulator is finished the project partner GfS will continue to run representative transients on the full-scope simulator to assess to sCO2-Loop performance.

With this delivery WP6 is well on track and a fully integrated coupled simulation model is available to further study the behaviour of a pressurized water reactor equipped with a sCO2-Loop.

3 Introduction

The transients run at the end of WP6 will be a main result of the sCO2-4-NPP project as they will show that the developed system will increase the safety of Nuclear Power Plants significantly by adding another safety system that ensures the decay heat removal during several accident scenarios.

To be able to achieve this result a Simulation Model of the sCO2-Loop needs to be connected to an existing full-scope training simulator.

In the following sections we will describe how this coupling was achieved using the interfacing mechanism described in deliverable D6.1.

We will start to explain once again the system architecture and the interface standard that was chosen to connect the models. In the following sections the reader will learn about the simulation boundaries and why the connections were placed at the agreed locations.

The remaining sections will describe all data points that need to be exchanged between the models to achieve a proper coupling. This section will be spilt in subsections that start with the data that needs to be exchanged for control purposes, continue with all thermodynamic properties that will be exchanged and finish with all data points that need to be transferred for display purposes. We will finish this section by explaining the tests we conducted to verify proper data exchange between the models.

After having described all interface mechanisms we will describe the hardware that was used to run the sCO2-Model and how the model is performing with respect to the real-time requirements.

Finally, the document will close by giving an outlook on the remaining tasks of WP6 which is the performance of transient simulation runs to prove the functionality of the integrated system.

4 Description of the interface mechanism

As introduced above we will now describe in detail how the code coupling will be performed and what achievements have been made so far in WP 6. The three basic requirements for software to be coupled to a real-time training simulator are:

1. Timing – The timesteps of the coupled software must be controllable to run the models synchronized. That means that the software must have an interface that allows the controlling simulator executive to request an evolvement of the model for a given time. It is crucial that the coupled software will finish its calculation within this given time. E.g., if the controlling simulation requests the coupled software to execute a simulation of 0.25 seconds, this time step must be finished in under 0.25 seconds. Only then real time behavior is achievable. It is very crucial for a training simulator to run in real time.

2. Initial Conditions – There needs to be a mechanism that the state of a simulation can be saved to disk at any given time. This is important to be able to restart the simulation from this saved state. This requirement is also crucial as the simulator cannot always be operated to a dedicated condition but needs to be able to load a previously saved condition.

3. Data exchange – This is probably the most straight-forward requirement for coupling simulation models. The two models need to have the possibility to exchange data points otherwise the coupled codes would have no chance to interact with each other. The crucial part of the data exchange is also that it must be executed in each time frame in order to keep the real-time requirements.

4.1 Interface architecture

The project partners from CVR will use the software package Dymola to model the sCO2-Loop in WP5. Therefore, this software is the limiting factor to choose a suitable interface architecture for the code coupling. As a first step CVR and KSG checked the options on coupling the third-party tools that Dymola provides.

The common standard to distribute Dymola models is the Functional Mock-up Interface Standard (https://fmi-standard.org/). The Functional Mock-up Interface (FMI) is a free standard that defines a container and an interface to exchange dynamic models using a combination of XML files, binaries and C code zipped into a single file (Functional Mockup Interface, 2021).

It was agreed between the partners to use the FMI Standard as the mechanism to interface to the Dymola model that will be provided by CVR. CVR will export the developed model to a so-called functional mock-up unit (FMU) and KSG will import the FMU into the simulator.

The standard supports all necessary functions to couple the Dymola model to the training simulator. The provided FMU can be loaded into the training simulator. It will expose all interface variables via API functions. The control of the FMU will be done via the simulator executive which can evolve the model in time by calling the respective FMU function. Using this function, the basic requirement of controlling the execution time of the coupled model is achieved and both models can run synchronized.

The FMI standard also supports serialization of a given state of the FMU i.e., the state of the FMU will be saved to hard disk. This state can later be loaded back into the simulation. Therefore, the second basic functionality of saving simulation states, so called initial conditions (ICs), and restoring the ICs is achieved.

The data exchange between the simulator and the FMU has been successfully verified on both demo FMUs that come with the FMI standard and the test FMU provided by CVR.

The training simulator that is used to run the transients, is operated in a Linux Red Hat operating system. The execution environment of the training simulator provides a TCP/IP based interface that allows communication with external models running in a Windows or Linux environment. The FMU provided by CVR can run on Windows as well as Linux operating systems. For the simulation, the Windows version of the FMU is used.

4.2   Simulation Boundaries

After having agreed on using the FMI standard to couple the two codes the WP partners had to decide on the modelling boundaries, i.e., where the models are coupled from a thermodynamic standpoint.
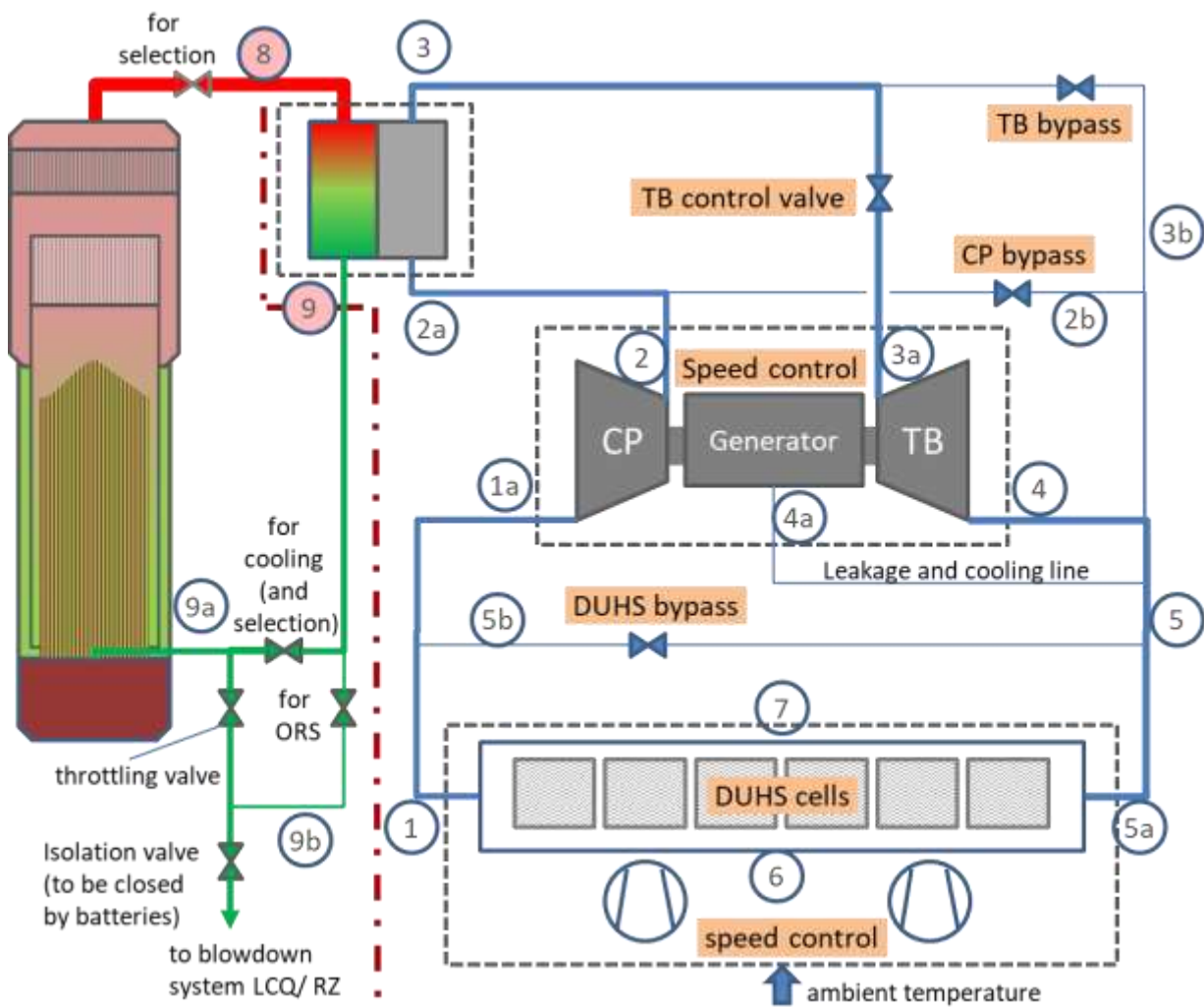

Figure 1: sCO2-Loop overview

Figure 1 shows the principle set up of the sCO2-Loop and how it will be connected to a steam generator in a pressurized water reactor. One can see that the connection points are at the main steam line (point 8) to the input of the compact heat exchanger (CHX) and that the condensate will be fed back into the steam generator blowdown line (point 9).

It was agreed between CVR and KSG to couple the codes via a fluid interface.

4.2.1   Coupling via fluid interface

It is a commonly used way to couple thermodynamic models via fluid interfaces. One side of the models provides basically a pressure and the thermodynamic properties of the fluid at the interface and reacts on the flow that is withdrawn from the other side of the model. Therefore, this coupling always consists of a pressure interface and a flow interface. To get a numerically stable situation a flow path always has a flow interface on one end and a pressure interface at the other. The connection in principle is shown in Figure 2.
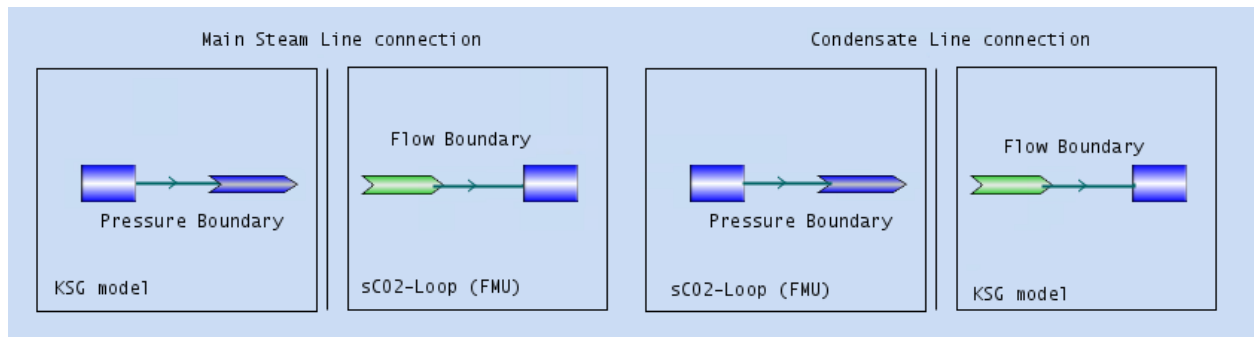

Figure 2: Interfacing scheme of pressure and flow boundaries

This coupling mechanism was chosen and agreed upon between the partners of WP6. Therefore, the model boundaries were drawn as indicated by the dotted red line in Figure 1.

The interfaces were added in the simulation networks of the training simulator as seen Figure 3 and Figure 4 marked by the yellow bounded boxes.

The final version had the pressure and flow interface switched compared to the initial report in deliverable 6.1. KSG will provide the main steam flow via a flow interface. CVR will provide the condensate properties via a pressure interface.
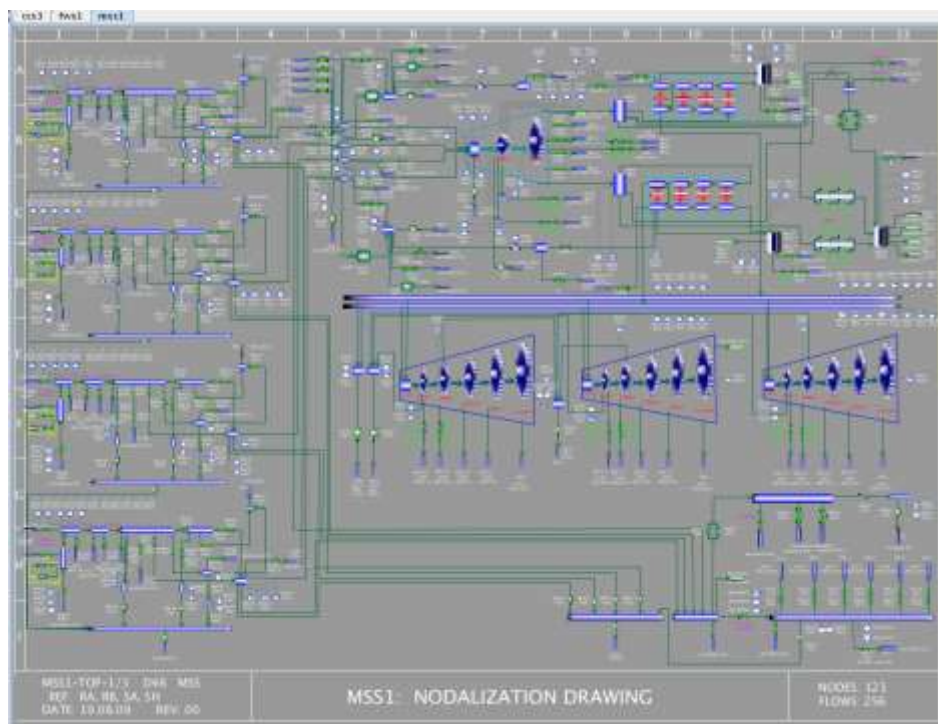
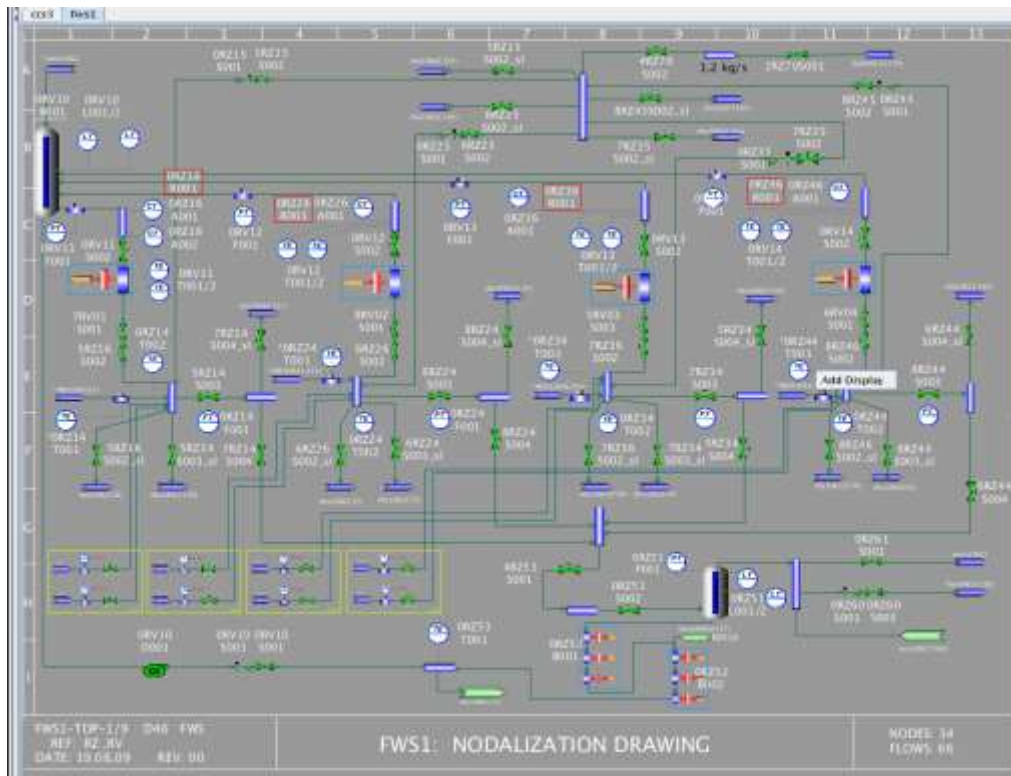**Figure 3: Main steam nodalization drawing**

Figure 4: Steam generator Blowdown nodalization drawing

4.3   Data exchange

As the interface architecture and the model boundaries have been agreed the remaining part to specify is the data set that has to be exchanged between the two models. This will be described in the following sections.

4.3.1   Command interfaces

As the sCO2-Loop is a self-contained, self-running, system there will be very limited interface points needed for command respectively control purposes of the model. All simulator functions are handled via the FMI-Standard. The control strategy of the sCO2-Loop is internal to the provided model and was not yet determined before the due date of this delivery. KSG will provide all necessary status values of the plant that are needed from the sCO2-Loop controllers.

Nevertheless, it should be possible to operate the sCO2-Loop in a manual or partially manual mode by the simulator instructor and to influence the sCO2-Loop operation. Therefore, the following possibilities will be implemented:

Table 1 Command interfaces

Command interfaces
Operation of all valves in manual mode
Adjustment of turbine speed setpoint
Adjustment of fan speeds
Select number of active DUHS cells

Adjust total mass of CO_2 filled into the system
Adjust remaining battery capacity available
Set efficiency for reload of batteries from generator
Set dirt or gas influence on heat transfer in CHX and DUHS (0…100%)

This limited set of command parameters allows the simulator instructor to manually operate the system and to set boundary conditions.

With the use of the FMI-standard it is easy to add command interface values during the implementation. This will be likely as the control strategy and the necessary measurements and input signals will be determined in the near future.

4.3.2   Thermodynamic interface variables

The coupling mechanism was described in 4.2.1. The thermodynamic code used by the KSG simulation is Topmeret which is a part of the JADE Toolset of GSE systems Inc. It is a two-phase six equation non-equilibrium model which solves the thermodynamic network in a fixed time step. The parameters described here are the standard set Topmeret uses to interface to other models or to itself.

The necessary interface variables for a pressure boundary are:

Table 2 Pressure boundary variables

Property	Unit
Node pressure	[bar]
Mass of gas/steam and liquid	[kg]
Density of gas/steam and liquid	[kg/m^3]
Enthalpy of gas/steam and liquid	[kJ/kg]
Gas/Steam void fraction	[dmnls[1]]
Concentration non-condensable gases	[dmnls]
Conductivity	[dmnls]

With this set of parameters, a pressure boundary provides all necessary information to calculate a mass flow that will be withdrawn from the connected flow boundary.

The necessary interface variables for a flow boundary are:

Table 3 Flow boundary variables

Property	Unit
Temperature of gas/steam and liquid	[°C]
Mass flow of gas/steam and liquid	[kg/s]
Enthalpy of gas/steam and liquid	[kJ/kg]

---

[1] dimensionless

Concentration non-condensable gases	[dmnls]
Conductivity	[dmnls]

With this set of parameters, a flow boundary provides all necessary information to calculate the pressure in a connected pressure boundary.

In case certain parameters are not in scope of the sCO2-Loop model, e.g., non-condensable gases or conductivity of the liquid phase, they can be set to default values to fulfill the interface requirements of Topmeret.

4.3.3   Display variables

As the FMU provided by CVR will run as an encapsulated module the only way to access internal variables is to provide them in the interface. In order to be able to monitor the behaviour of the sCO2-loop it is therefore necessary to have a certain set of display variables as output of the FMU. KSG will setup a graphic to visualize the operational state of the sCO2-Loop. This graphic will look similar to Figure 1. On all 19 points of interest (1 through 9b) the following parameters will be displayed:

Table 4 Display parameters

Property	Unit
pressure	[bar]
temperature	[°C]
mass flow	[kg/s]
density	[kg/m^3]
enthalpy	[kJ/kg]
entropy	[kJ/K]
speed of sound	[m/s]

Having access to these operational parameters will give a profound overview of the state of the sCO2-loop. In addition, the following internal parameters are of interest:

Table 5 Internal Parameters

Internal Parameter	
speed of fans	surplus electrical power
speed of turbine	work of turbine
position of valves	work of compressor
CO_2 mass in vessel (CHX, DUHS)	efficiency of TB
number of DUHS cells active	efficiency of CP
heat removed via CHX	distance to surge line (Mach numbers)
water mass in CHX	status of battery power supply
electrical power from/ to generator	battery capacity remaining
electrical power for fans	

Again, it will be a straight-forward process to add display variables to the interface when during the project the above list is deemed to be incomplete. Figure 5 shows the overview picture of all steam generators and all sCO2-Loops. It can be utilized to quickly get an understanding of the system behavior during transients.
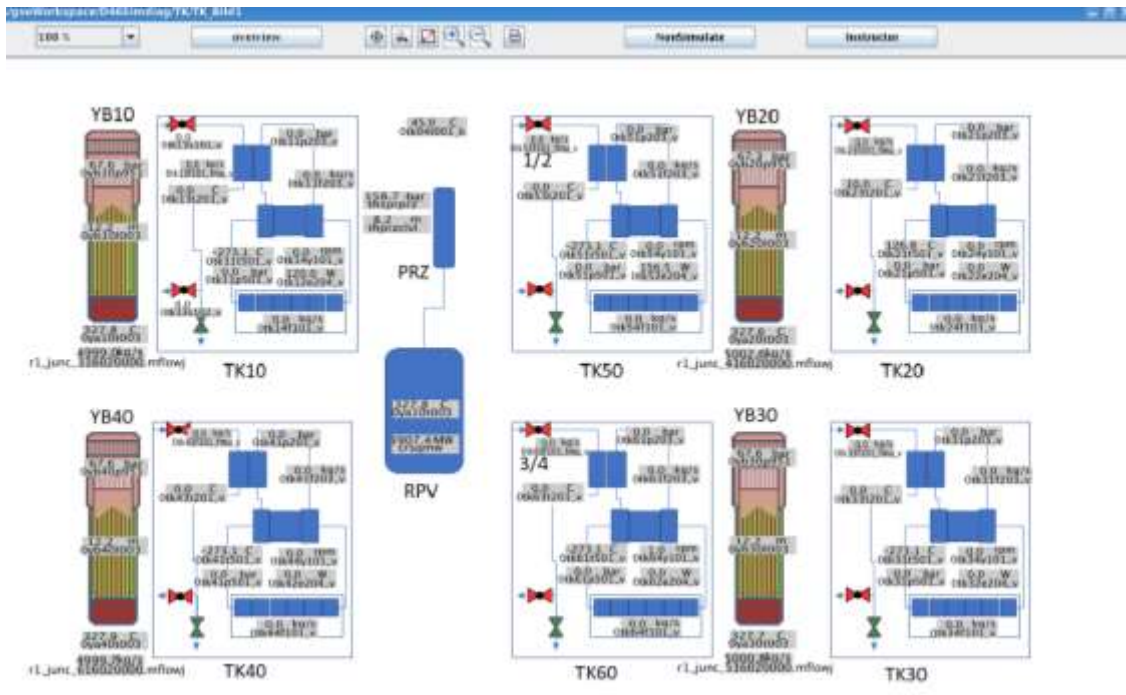

Figure 5: Overview of Steam generators and sCO2-Loops

To check and operate one specific sCO2-Loop the schematic views shown in Figure 6 are used. The instructor can manipulate all components of the sCO2-Loop via the controls on this diagram. The actual loop parameters are displayed on the specific measurements. Navigation buttons are introduced on the left-hand side to switch between all available displays.
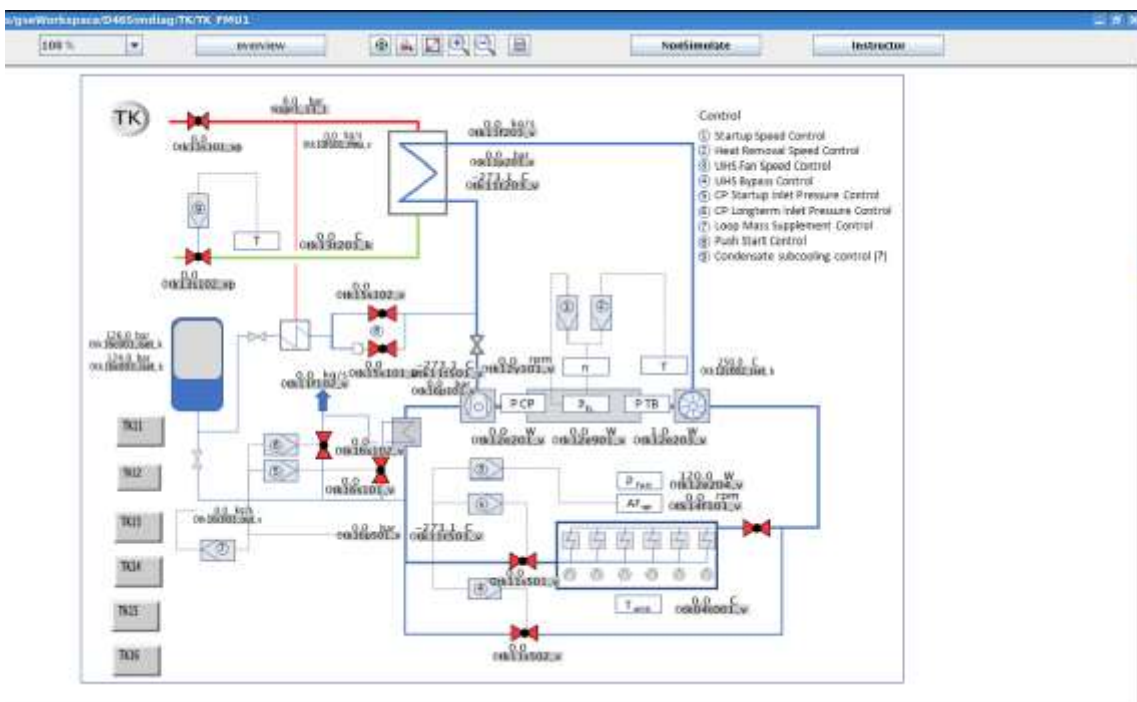

Figure 6: Schematic view of a specific sCO2-Loop

So, with the above interface description all necessary parameters that could be determined at this stage of the project have been fixed.

4.4   Hardware

To execute the sCO2-Model FMUs a dedicated virtual machine was set up in the KSG datacenter EKu.LOC. The virtual machine has the following hardware specification.

Table 6 Hardware Specification FMU-Server

Hardware Specification	
CPU	6 Cores, Xeon Gold 6240Y CPU @ 2,59 GHz
Memory	16 GB
Disk	250 GB
Network	1x Gbit/s

The VM was connected to the simulator network via VLAN. This proves that the communication between the FMU-server and the FMU-client can work via TCP/IP-interface.

The hardware performance during steady-state operation was evaluated. Network traffic is neglectable and CPU power is operating around 80 %.

During first transient tests it was proven that the chosen hardware setup can run the sCO2-Model in real-time during most operations. Real-time behavior of the Model will be further investigated in the upcoming testing period.

4.5   Outlook

Having defined the interface architecture and the interface variables CVR did already provide a working FMU to KSG. This FMU has already been connected successfully to the training simulator which has proven the chosen interface mechanism to work. With the close collaboration between CVR and KSG it was achieved that the real-time requirements are met with the current modelling state. The delivered FMU was successfully integrated and GfS can now start running transient tests on the full-scope simulator.

5 Conclusion

The major achievement of Deliverable 6.2 is a coupled simulation system between the KSG training simulator environment and the Dymola sCO2-Loop model of CVR. The interface mechanism defined in Deliverable 6.1 has been implemented on the full-scope simulator and the sCO2-FMU has been deployed on a dedicated virtual machine. Coupling of the models did work seamlessly due to the preparation work in the early WP6 phase.

Therefore, a setup was established to demonstrate the capabilities of the sCO2 technology in a real nuclear power plant simulator.