Integrated building and system simulation using run-time coupled distributed models
Trcka, M.; Hensen, J.L.M.; Wijsman, A.J.T.M.

Published in:

Published: 01/01/2006

Document Version
Accepted manuscript including changes made at the peer-review stage

Please check the document version of this publication:

• A submitted manuscript is the author's version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

Citation for published version (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal ?

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 15. Dec. 2018
Integrated Building and System Simulation Using Run-Time Coupled Distributed Models

ABSTRACT

In modeling and simulation of real building and heating, ventilation and air-conditioning (HVAC) system configurations, it is frequently found that certain parts can be represented in one simulation software, while models for other parts of the configuration are only available in other software. However, very often quality assurance dictates that the configuration should be modeled in an integrated fashion due to thermodynamic and other interactions between the various subsystems.

This paper gives an overview and presents initial results of an ongoing research project that aims to improve the applicability and use in a design context of building performance simulation software by addressing this problem. The objective of the work is to research, enable and validate run-time external coupling of distributed building and system simulation software. Although the research uses specific simulation environments, the coupling mechanism(s), data-exchange procedure and protocol(s) that are being developed ensure that the approach has a wide and general applicability.

This paper also discusses some of the domain-specific theoretical and numerical issues regarding run-time interoperability and parallel coupling of distributed HVAC component models. A first prototype is presented. The initial results for a practical case study are used to demonstrate the prototype and to illustrate the usefulness of the approach. The paper concludes with indicating directions for future work.

INTRODUCTION

Integrated building performance simulation can help in reducing emission of greenhouse gasses and providing substantial improvements in fuel consumption and comfort levels. This is done by treating buildings and the systems which service them as complete optimized entities and not as the sum of a number of separately designed and optimized subsystems or components. However, due to the decentralized nature of research and developments in this area, available components are distributed between different environments. Certain component models are available in one simulation package, while other components are only represented in another.

It has previously (Hensen 1991; Hensen and Clarke 2000) been argued that in the area of system simulation there is still enormous amount of work to be done. System modeling and simulation capabilities develop very slowly and take up an enormous amount of resources (time-wise and financial). An efficient way forward would be to share developments and to reuse existing software.

The work underlying this paper started with literature review and investigation of possible strategies for sharing developments. An overview of available techniques is given in (Hensen et al. 2004). For example, data model interoperability may be achieved on the product model level, either by sharing (Lockley et al. 1994) or exchanging (Bazjanac and Crawley 1999) information. Even though a common product definition model eases the use of simulation tools, it addresses only part of the overall problem.

Additional interoperability may be achieved on the level of physical process models. This can be realized on source code level or in a more generic way by expressing the models in a neutral format, such as NMF (Bring et al. 1999) and Modelica (Tiller, 2001). Both data and process model interoperability take place before run (or execution) time, as shown in the upper part of Figure 1.

This paper introduces the concepts and core issues of an approach to process and to data cooperation by run-time external coupling of distributed simulation environments. As indicated in Figure 1, the work does not address model definition interoperation, nor does it consider sharing result sets from the coupled programs.
The research method is based on rapid software prototyping. Subsequent prototypes are developed on the basis of implementation experiences and simulation results for various realistic case studies. The case studies also serve for proof-of-concept and to demonstrate the applicability of the external coupling approach by highlighting its potentials.

The present prototypes are based on an advanced building energy simulation program (ESRU 2000) that has been extensively validated. Although the prototypes are implemented in the particular simulation environments, further prototype testing within a broader range of simulation packages will ensure that the obtained results and experiences are software-independent in order to guarantee that the approach has a wide and general applicability.

This paper shows the first results of this ongoing work. The issues associated with implementations of run-time coupling are discussed. Finally, early prototypes are presented and demonstration results are shown.

EXTERNAL RUN TIME COUPLING

External run-time coupling addresses the case where at least two programs are executed simultaneously and where information (i.e. simulation results) is exchanged between them. Previous and ongoing work by others in this area includes coupling of lighting and building energy simulation (Janak 1997), CFD and building energy simulation (Djunaedy et al. 2003; Zhai 2003) and between building energy simulation and system simulation (TESS 2003; AIGUASOL 2003; CSTB 2003; Curtil 2004).

Potential advantages of external run-time coupling are:

- There is much greater flexibility in HVAC system modeling and simulation.
- It allows coupled programs to run on separate and even different computers in parallel.
- New developments in coupled component models can be automatically included, i.e. it is not necessary to update the base program or any of the other coupled programs.
- External programs may be treated as black boxes, so other parties can join and share their models without giving away the know-how.
- If real-time simulation is enabled, and as every component/group of components is treated as an independent object, it is not difficult to imagine that besides connecting external software package, an external piece of hardware can be connected as well, thus enabling hardware-in-the-loop testing.

One of the main challenges is that the variety of models and tools, together with implementations of different coupling strategies, could influence the overall solution. There are also many questions to be answered. Which data should be exchanged? What should be the frequency of data exchange? How do different strategies influence the overall solution and its stability? This paper gives an overview of such issues and presents possible implementation strategies.

EARLY PROTOTYPES

Terminology

In this paper the term base program is used to address the building energy simulation program, which also functions as the overall simulation controller. (A separate overall simulation controller may be considered in the future.) The term external program is used to address the program that simulates components of the overall building and system configuration that cannot be modeled in the base program. Until now, two implementation approaches have been researched.

Mechanism with discontinuous external program run

In this implementation, the base program invokes the other application when necessary and coordinates the information exchange between them. Currently, intermediate files are used for data exchange, but any other inter-process communication (IPC) mechanism (Yahiaoui et al. 2003) could be used as well. During the simulation run-time, the base program invokes and prepares and sends the necessary data to the external program. While the external program does its calculations, the base program waits. After completion, the external program sends the relevant simulation results back. The base program then continues its calculations. The general procedure is schematically shown in Figure 2.

The data are exchanged in both directions. The mass flow rate, temperature (second phase mass flow rate can be included) and optionally control variable are sent from base to the external program. The
calculated temperature and/or mass flow from external program are forwarded back to the base. Additionally, the working model incorporates an option for a user definition of frequency at which the external program is invoked. The frequency may differ from building and system calculation frequency in the base environment. However, it has to be set up according to the simulation time or time step of the external program.

The external program is invoked each specified time step from the base program. While the base program continuously runs, the external program starts and stops during the overall simulation time. To keep the dynamic evolution of the results, the necessary information is externalized. To initialize the external (history) data, some preprocessing – meaning performance of the pre-calculation with the external program – is needed. The transfer of the history data between different time steps can be seen as an artificial continuous program-run. Later in the paper this mechanism will be addressed simply by discontinuous mechanism.

**Mechanism with continuous external program run**

In this implementation, the coupled programs are invoked and running simultaneously. Each coupled program needs an interface for data exchange. This implementation is made to work with so-called “named pipes”, but other IPC mechanisms could be used as well. Named pipes are an inter process communication (IPC) protocol for exchanging information between two applications. Process blocking and synchronization tools are provided as part of the pipe services. Read and write operations to a named pipe are blocking, by default. If a process reads from a named pipe and if the pipe does not have data in it, the reading process will be blocked. Similarly if a process tries to write to a named pipe that has no reader, the writing process gets blocked until another process opens the named pipe for reading.

The programs execute synchronously in parallel and exchange data at a user-defined frequency (Figure 3). Two named pipes are used, one for each direction of data exchange.

This implementation does not need additional history data management, but adaptation of the external component model is still necessary. Besides redirection of input and output data, the external program will have to be able to send and receive the data at the specific frequency during the run-time.

Later in the paper this mechanism will be addressed simply by continuous mechanism.

**DISCUSSION**

**Data to-be-exchanged**

In order for the external coupling to work properly, it is necessary that relevant information is communicated between the coupled programs. One important aspect is which physical and numerical variables will be exchanged between components models in the coupled environments. The data should, as much as possible, represent physical quantities as they could be measured in the real world; i.e. as opposed to derived or abstract variables.

It could be argued that except control components, all other HVAC components “communicate” via working fluids such as water and moist air. This is the starting point for inter-component communication here. The main advantages of this are:

- Since the data represents physical quantities, it should be readily available in any software program.
- Since such data can also be readily obtained from building energy management systems and other data-acquisition systems, it would be relatively easy to enable run-time coupling of the simulation environment with a real building (e.g. for control purposes) or with system components in a test-rig (e.g. for hardware-in-the-loop testing).

The above also implies that by choosing a minimum set of variables that determine the thermodynamic state of the fluid, plus a variable that quantifies the flow, the complete information can be communicated from one to the other. We assume that the communicated variables are expressed in SI units. Some additional, but trivial transformation and mapping might be necessary if the coupled program uses a different system.

First, the minimal number of variables that determines the thermodynamic state of the fluid is defined. This can be done with the Gibbs phase rule: 

$$p + f = c + 2$$

where: 

- $p$ is the number of phases,
- $f$ is the variance or number of degrees of freedom in the system, and
- $c$ is the number of fluid components.

For example for water: $p=1$, $c=1$ lead to two degrees of freedom. Thus, two state variables are needed to
determine the state of the water. In the case of moist air, \( c=2 \) and the number of degrees of freedom equals 3. Minimum number of variables to be exchanged is defined as number of degrees of freedom plus a variable that quantifies the flow.

When the minimum number of variables is theoretically known, their quality should be defined. This subject has been addressed many times before. In many HVAC component modeling approaches, pressure drop is neglected, and mass flow is considered to be pressure independent (Hensen 1991; Bring et al. 1999). This avoids having to define the exact pressure loss for each component. In relevant cases, pressure head may be used to estimate the power of pumps and fans. Thus, the pressure may be excluded as a relevant state variable, and the minimum number of variables decreases by one. For two degrees of freedom, Sahlin and Sowell (1989) define mass flow rate and enthalpy as the variables involved in interactions between components. If the number of degree of freedom increases, the additional variable holding the information about the humidity (moist air) has to be included. For that case Sowell and Moshier (1995) list the most common set of interface variables:

- volume flow rate, temperature and relative humidity
- mass flow rate of dry air, temperature and humidity ratio
- mass flow rate of dry air, enthalpy and humidity ratio

The second set of variables was preferred on the bases of directness of their applicability in the conservation equations and their familiarity to practitioners.

Until now, one additional problematic issue has been implicitly introduced. Component models in different environments require different sets of input variables. Additionally, different standard units as well as dimensions are possible. Our approach aims to enable the use of “all” available HVAC components models without any limitations, and a certain mechanism of conversion has to be established and implemented. An idea that was initiated in Sahlin et al. (1995) and later developed Sowell and Moshier (1995) and Sowell et al. (2001) of so-called auto-link could be used in the context of external coupling. The link here carries the information of fluid flow rate and the set of state variables that define its thermodynamic state (minimum number), plus set of equations that describes the relations between the properties. The equations cover relations regarding units as well as different thermodynamic properties.

However, as indicated above, in the developed prototypes only SI units are exchanged with mapping and transformation in the component models if applicable.

Besides working fluid coupling, there might be a need for exchanging control information among the programs, as the sensor and actuator part of a control loop may not be simulated in the same environment. Therefore, the option for exchanging control data is also enabled and can be used optionally.

**Coupling strategies relative to different component models**

In the run-time coupling approach, each application runs separately, while they interact through their boundaries. There are two different external run-time coupling strategies:

1. Quasi-dynamic coupling (Zhai 2003), or loose coupling (Struler et al. 2000), or ping-pong coupling (Hensen 1999).
2. Fully-dynamic (Zhai 2003), or strong coupling (Struler et al. 2000), or onion coupling (Hensen 1999).

Accuracy as well as stability constraints limit the simulation time step length in case of the first strategy. The second strategy allows longer time steps for the same accuracy, but it requires an iteration procedure to ascertain user-defined convergence criteria. A potential problem that is associated with the second strategy may arise from divergence or stagnation of convergence across multiple applications (Struler et al. 2000). The following discusses the above strategies in relation to different component models and implementation mechanisms.

In the case where the external component is modeled in steady state manner, there is no need for history data. Both quasi- and fully dynamic coupling strategies can be used in combination with both discontinuous and continuous implementation mechanisms. It is our opinion, in this early stage that discontinuous mechanism is more appropriate as it does not require synchronization.

It is a different case for the coupling to a transient component model, where history data plays an important role. The fully dynamic coupling would cause problems associated with synchronization and history data management if used in combination with these types of models. In the first iteration the history data set up initial conditions. At the end of the calculation in the first iteration, new history data are memorized in order to be used in the next time step. If another iteration is needed, the history data available
are not the correct data anymore, as they were overwritten by the data from the previous iteration. It is obvious that iterations require the *rewind* of the external component as well as additional data to be exchanged for the rewinding control. This appears to be a complex and unpractical approach and not exactly a good alternative to the more traditional ones discussed in the introduction. Therefore, a preferred option for the external coupling of transient models is shortening of time steps rather than applying iterations. Hence, the quasi-dynamic coupling strategy is favored together with the continuous coupling mechanism that does no require externalization of history data.

Some issues concerning external coupling of simulation environments that employ different system solution approaches (input-output based and conservation equation based) are discussed elsewhere (Radošević et al. 2005).

**Time stepping issues and accuracy**

Time constants of coupled components potentially range from minutes to months. Thus, there may be different requirements for maximum time steps in the coupled programs, as indicated in Figure 4.

If the time constant of the external component is shorter and consequently requires smaller time steps, the same shorter time step should be used in the base program. This is indicated by the dashed line in Figure 4. The specific dynamics of the external component will not be lost and it will evolve with the input variables’ changes in time. It could be possible to apply different time steps between base and external program, but then dynamics of the external component would be neglected and the component would be treated as steady state one from the perspective of the base program. This may have sense only if the particular dynamics do not influence the behavior of rest of the system.

On the other side, if an external component has a much larger time constant then the rest of the system (a heat storage, for example), the multi-time-step approach could be applied, while the maximum allowed time step should be determined by accuracy and maybe stability constrains and the time-scale of relevant system parts.

In Belytschko et al. (1985), the authors developed stability conditions for multi-time-step coupling of mixed – both explicit and implicit – solvers. They found that either implementing implicit numerical schemas or limiting time steps in both coupled environments (or combination of both) would give a stable solution. Accuracy will be, however, a function of the length of a time step. They did not mention the influence of nonlinearities nor the influence of time-lagging coupled information, which should not be neglected. The stability of similar problems was addressed in Larsson (2001), who reports that “if only straightforward exchange of state-variables is performed among the solvers, the coupling may generate numerical problems.” One of the options to improve the coupling is as stated in Elliott (2000), by using (linear or) quadratic functions for extrapolation and interpolation of the exchanged variables.

There are numerous options of how the data could be exchanged, depending on which values from which time steps are used, as well as when the values should be updated and should they be kept constant or linearly extrapolated (Grott et al. 2003) or interpolated (Gravouil and Combescure 2001) in time. There is no a straightforward solution, and the aim in the current work is to study various approaches and their impact on the overall solution. Investigation of stability issues will be the next step in the research.

An additional feature could be added to the external coupling mechanism such as *freezing* the external component. This means that if nothing changes in the system, the output values of the external component are held constant and equal the values from the previous run until some condition is fulfilled when the data should continue to be exchanged. How to pose the criterion for freezing and how to combine it with different coupling mechanisms is an issue still to be investigated.

**DEMONSTRATION CASE**

To test the prototypes, a low energy building in Korea (Yoon et al. 1997) was used (Figure 5). It includes an earth-to-air heat exchanger for preheating or precooling fresh air supply, depending on the season. The building incorporates a double-skin south facade, which can act as a solar collector for additional preheating of supply air. In the summer this is bypassed to avoid overheating of the air. In that case the double-skin facade is naturally ventilated. The transient solar behavior and the dynamic nature of the double-skin facade and the ground-coupled heat exchanger affect the building heating and cooling load in a very dynamic way.
As indicated in Figure 5, the building, including the double-skin façade, is modeled in one simulation program, and the earth-to-air heat exchanger is modeled in another. Both simulation programs have undergone minimal code adaptation.

The tested working prototype employs a continuous external program run mechanism and quasi-dynamic coupling, i.e. no iterations.

**Inter-program time step variable data exchange**

Figures 6 and 7 schematically show inter-program time step variable data exchange for programs used in this particular case. For calculation at the certain simulated time, t, the coupling data are sent to the external program. The data for the time t, are yet unknown and the data from the previous time step are used. The external program performs the simulation for the specified time step and sends the updated coupling data back to the base program. There are many possibilities as to which data to send back. They can be either the last calculated data (Figure 6) or the mean over the particular time step (Figure 7) when time steps among the programs differ, or any similar combination.

To test the difference in possible combinations in inter-program time-step variable data exchange, we performed an experiment. The building with the double skin was simulated with a time step of one hour, while the coupling was performed less frequently, i.e., with the time step of several hours. Although the time constant of the external component is not concidelably larger then the time constant of the rest of the system, we found it interesting to observe the impact of the various coupling combinations on the results.

Three simulations were performed: (1) with the coupling time step (of one hour) equal to programs’ simulation time steps and (2) implementing coupling time step of three hours, while keeping the programs’ time step at one hour. For the latter case, coupling variable was equal to the last calculated data in one simulation and to the arithmetic mean value for the particular coupling time step in the other simulation. The results are shown on Figure 8.

Although it appeared logical not to use the last calculated value but the mean over the particular coupling time step, the experiment showed the opposite. The coupling of the mean value over the coupling time step results in incorrect coupling data values by decreasing the rate of change of the coupled variable.

Thus, the last calculated value of a coupled variable is chosen to be exchanged between programs.

**Necessity of the coupling approach**

The demonstration case represented here does not exploit the full potential of external coupling approach, as the simulated system is a purely open system, i.e., system components modeled in different programs do not require each others feedback. Therefore, the model could be decoupled. The decoupled solution means that the output of one program will be redirected to the input to the other in two subsequent simulation procedures. However, we performed coupled simulation by making the ground-coupled exchanger a part of the overall system, as shown on Figure 5.

The demonstration case involves comparison of the energy-saving potential of several design options of the earth-to-air heat exchanger coupled to the double-skin façade. The simulations assume Korean climate and a period of one winter week. Additionally, the results are compared to those obtained from the approach used in Yoon et al. (1997). There, the model uses monthly constant value for the temperature of the air entering at the bottom of the double-skin façade. It was estimated that this temperature is equal to the ground temperature, evaluated in Equation 1.

\[ T_{gl} = T_{a,j} + 0.5(T_{m,j} - T_{a,j}) \]  

(1)

where:

- \( T_{gl} \) = monthly mean ground temperature;
- \( T_{a,j} \) = annual mean ambient dry bulb temperature and
- \( T_{m,j} \) = monthly mean ambient dry bulb temperature (Yoon et al. 1997)

Based on recommended velocities in literature (Mihalakakou et al. 1996; Mihalakakou 2003; Pfafferott 2003) the pipe radius is determined by the design volume flow rates. Different design options are simulated with two volume flow rates: (1) lower - 0.25m³/s would be sufficient for ventilation of the building, and (2) higher - 1m³/s to compare the coupled to uncoupled approach. This resulted in pipe velocities between 2 and 8 m/s. The pipe depth was kept between 1.5 and 3 m and the pipe length between 30 and 150 m. A one-
pipe heat exchanger was used in case of the lower volume flow rate, while four parallel tubes were considered for the higher volume flow rate (for details see Tables 1).

The gross heat gain by both the ground-coupled heat exchanger and the double skin is evaluated from the difference between the ambient temperature and the temperature at the upper part of the double skin. It was assumed that system operates between 8 a.m. and 19 p.m. In some hours the heat gain is higher than the overall loss of the building itself. The minimum of these two values at each point in time were used to assess the energy-saving potential. The results are shown in Figure 9 and 10.

| TABLE 1. Design parameters for different design options and lower volume flow rate |
|-------------------------------|------------|--------------|
|                               | Design1    | Design2      | Design3      |
| Depth [m]                     | 1.5        | 2.5          | 3.0          |
| Length [m]                    | 30.0       | 70.0         | 150.0        |
| Radius [m]                    | 0.2        | 0.11         | 0.12         |

The temperature difference of the incoming air at the bottom of the double skin does not have significant influence on the overall result in the first case (Figure 9). With the lower volume flow rate, the ventilation loads have less impact on the resulting temperature of the double skin compared to solar heat gain and loads due to conduction through the construction. However, if the volume flow rate is increased, (Figure 10) the temperature difference of incoming air will have much more impact on the simulation results.

It can be concluded that for low volume flow rates, the simplified model reasonably well predicts the energy-saving potential (in this specific case the difference is less then 10%). In this particular case the incoming temperature, hence, earth-to-air heat exchanger itself, does not have a big influence on the results. However, the simplified, uncoupled approach does not allow evaluating the influence of different design options when the volume flow rate is higher. The deviation here rises up to 25%. The coupled approach will be necessary to predict the energy-saving potential in this case.

CONCLUSIONS

The external run-time coupling approach promises to be very flexible in several respects. Current limitations of non-shared developments in HVAC component modeling can be easily overcome as the various parts of a building with systems configuration can be simulated in different environments. Additionally, components with different dynamics may be simulated with different time steps. This paper focuses on discussing issues involved in the development of the coupling itself, such as which information needs to be exchanged and at which time intervals.

Depending on the specific nature of a component itself the coupling philosophy is derived as follows:

1) For all steady state component models, the discontinuous mechanism seems to be the most appropriate. There is no need for history management and iterations can be easily applied.

2) In the case of coupling to transient component model with a time constant smaller than those of the rest of the system in order to obtain accurate and stable results, the time step of the base program should be shortened to the value necessary to perform calculations of the external program. This issue will be investigated further in the near future.

4) If the model of external component model is transient and the external program has a time constant that is similar to one of the base program, the time step should be defined in a way to avoid iterations. If this is not possible, a database of history data should be maintained. Both coupling mechanisms can be used. If discontinuous is used, the properties relevant for dynamic behavior of the component should be externalized. However, this can be avoided by using the continuous approach.

5) For the transient external component model that has a time constant larger than the one of the base program, the continuous mechanism seems to be the most appropriate. Multi-time-step approach can be
applied, but in order to avoid the need for iterations, some limitation criteria should be posed to the maximum allowed time step.

FUTURE WORK

Immediate future work will include further development and testing of the prototypes concerning stability issues. Various other building software tools will be used to advance the prototype and broaden its applicability. The co-operative building and system simulation environment will be verified using one or more of the known validation techniques: comparison with physical experiments performed under controlled laboratory conditions, comparison with analytical solution, or comparison with results from other programs. The practical verification will be based on the application of at least two real design studies.

Associated with this will be the generation of guidelines regarding the necessity/applicability of building and system simulation and the cooperative approach in terms of integrated design of buildings and systems.

REFERENCES


Curtil D. 2004. Personal Communication


LIST OF CAPTIONS FOR FIGURES

Figure 1
External run-time coupling in the context of other interoperation research and developments

Figure 2
Flow chart – Mechanism with discontinuous external program run

Figure 3
Flow chart – Mechanism with continuous external program run

Figure 4
Comparison of different time steps

Figure 5
Case study for testing the first prototype couplings

Figure 6
Inter-program time step variable data exchange; Coupling time step equal to individual time steps within programs

Figure 7
Inter-program time step variable data exchange; Coupling time step equal to a larger time step

Figure 8
Coupling temperatures for various coupling time steps and different combinations of the coupling variable

Figure 9
Estimated weekly energy saving for several design option and lower volume flow rate, including the result for the uncoupled simple earth-to-air modeling approach

Figure 10
Estimated weekly energy saving for several design options and higher volume flow rate, including the result for the uncoupled simple earth-to-air modeling approach
Figure 4
Figure 8

- **ambient temperature**
- **coupling temperature with coupling time step of 1h**
- **coupling temperature with coupling time step of 3h**
- **coupling temperature with coupling time step of 3h calculated as the average over the particular coupling time step**

Temperature [°C]

Time [h]
Comparison of coupled and uncoupled simulation for lower volume flow rate

Energy saving potential [KWh]

- without coupling: 380 KWh
- design1: 370 KWh
- design2: 360 KWh
- design3: 350 KWh
Comparison of coupled and uncoupled simulation for higher volume flow rate

![Comparison of coupled and uncoupled simulation for higher volume flow rate](image)

- **Energy saving potential [kWh]**
  - **without coupling**: 800 kWh
  - **design1**: 900 kWh
  - **design2**: 700 kWh
  - **design3**: 600 kWh