Analysis and Development of an Object-Oriented Library Structure for Solar Thermal Power Plants

Georg F. Schneider[1], Paul Gauché[2]

October 27, 2012

Abstract

CSP is recognized as a key technology to solve future energy scarcities mainly in countries with high solar irradiation. Prior to the capital intensive construction of Concentrating Solar Power (CSP) plants mathematical modeling is necessary for energy yield projection. However no standardized methodology is available yet. We present a library structure and basic models suitable to host standardized component models of CSP plants and show implementation examples. Our library structure shows that object-oriented modeling techniques are an appropriate way to assist standardizing of modeling approaches for CSP plants.

Keywords: Modelling; Simulation; Solar thermal; Latent heat storage; Object-oriented; Library structure; Concentrating solar power (CSP)

1. Introduction

One of the main questions which have to be answered in this century is how to satisfy the world's increasing energy demand. Stressing is the fact that fossil resources are constraint and human impact on world´s climate starts to be inevitable. Concentrating Solar Power (CSP) is recognized as a key technology to solve future energy scarcities mainly in countries with high solar irradiation. CSP has been identified as a primary long term method of generating sustainable power in South Africa.

Constructing CSP plants is still capital intensive therefore an extensive engineering process in plant development is necessary. During all stages of the process, mathematical modeling is applied to calculate the expected energy yield.

The SolarPACES project "Standardization of CSP Performance Model Projection", (STAMP) started in September 2010 and aims to develop, document and publish guidelines for CSP energy yield analysis with an international collaboration (Hirsch et al., 2010), (Eck et al., 2011). Within this project a structuring methodology for the division of CSP systems into subsystems is proposed (Eck, Benitez, Hirsch, Ho, & Wagner, 2010).

---

[1] Bsc (Mech), Visiting research student at Stellenbosch University, RWTH Aachen University, Templergraben 55, 52062 Aachen, Germany, Phone: +49 177 88 96 77 0, Email: Georg.Schneider@rwth-aachen.de

[2] Meng (Mech), Sr. Researcher and Director STERG, Dept. Mechanical and Mechatronic Engineering, Stellenbosch University, Private Bag X1, Matieland, 7602, South Africa, Phone: +27 21 808 4242, Email: paulgauche@sun.ac.za

The work presented in this paper shows how object-oriented techniques assist the development of an appropriate library structure to support further development. A library structure is presented and its feasibility is proven by implementing a component to determine the sun position and measured data using the simulation environment Flownex. Basic models are defined from which more complex models can be derived.

Nomenclature

Latin letters	
B	Value in **Equation 3-3** [degree]
E	Rotation speed correction according to **Equation 3-3** [min]
N	Day of the year [d]
t	Time [h]
Greek letters	
$\boldsymbol{\alpha}$	Azimuth angle of the sun [degree]
$\boldsymbol{\gamma}$	Elevation angle of the sun [degree]
$\boldsymbol{\delta}$	Declination of the sun [degree]
$\boldsymbol{\lambda}$	Longitude [degree]
$\boldsymbol{\varphi}$	Latitude of plant position [degree]
$\boldsymbol{\omega}$	Hour angle of the sun [degree]
Suffices	
loc	Legal (watch), local
sol	Solar
std	Standard of time zone
s	sun

2. Library Development and Object-oriented Analysis in Literature

Libraries for power plants and in particular solar thermal power plants have been developed in the past. In this review, due to cost constraints, only open source libraries have been investigated, leaving out the fact that several commercial model libraries are available.

Most known is the library for Solar Thermal Electric Components (STEC) developed to model and simulate solar thermal power plants in the software environment TRNSYS using the Isibat (nowadays Simulation Studio) Interface (Schwarzbözl, 2006). It is written as all TRNSYS types in FORTRAN subroutines following a procedural modelling paradigm. The library structure itself is partly structured according to the contributors to the project and no basic structuring guidelines are presented. Beside the bare source code for each component, the documentation is limited to very few models. Nevertheless a basic division into flow related and optical systems can be recognized and certain packages contain component models with differing detail.

The National Renewable Energy Laboratory (NREL) developed the software System Advisor Model (SAM) which is freely available (https://sam.nrel.gov/). It is a performance and financial model designed to facilitate decision making for people involved in the renewable energy industry. It contains several modules to investigate almost all renewable energy technologies. Due to this approach SAM is structured with an object-oriented programming paradigm using reusable components. For example all models in the CSP category use a power cycle model that consists of four major parts i.e. plant design, plant cooling mode, dispatch control and operation. If necessary, thermal storage can be specified and is taken into account. However, the underlying models of SAM are not accessible and modification or addition of models is restricted.

The ThermoPower library is a library written in the modelling language Modelica by Francesco Casella (Casella, 2003). The scope of the library is to offer support for the generation of power plant models giving modular, dynamic models and the level of detail is constraint to lumped and one dimensional models supporting the main goal to simulate power generation systems as a whole. The flat hierarchy of the library causes numerous classes at the top level, which makes it difficult to understand it quickly. However, the library does not contain any optical models for solar plants.

These examples do not fulfil the requirements for the definition of an international standard of CSP plant modelling. Therefore a complete new library structure was developed which connects the strengths of object-oriented modelling and a structured approach to divide the CSP plants into subsystems following the energy conversion through the system.

3. Object-Oriented Library Structure

3.1. Methodology of Library Development

Various authors have investigated methodologies to implement component libraries using the object-oriented approach ((Mühlthaler, 2001), (Eborn, 2001), (Tummescheit, 2002), (Casella & Leva, 2006) and (Richter, 2008)). In particular Richter, (2008) and Mühlthaler, (2001) worked intensively in their dissertations on giving guidelines and best-practice approaches for library development. The work of these two authors, as well as the authors experience was considered by designing the proposed library structure (see 3.2, p.- 4 -).

Richter, (2008) postulates six guidelines for library development (slightly modified):

(1) Class names should begin with an uppercase letter, object names with a lower case
(2) Inheritance should only be used for a is-a-relationship between two objects
(3) Multiple inheritance should be only used for relationships following guideline (2)
(4) Multiple inheritance should be avoided whenever possible
(5) The object-oriented structure of a component model library has to be constantly reviewed during the design process
(6) The Inheritance hierarchy should be as flat as possible

Mühlthaler, (2001) presents a more elementary work. He emphasizes the possibility of the components to be reusable in a direct meaning, or using specialization and polymorphism. He recommends structuring the library following the topology of the plant. As interface variables he recommends, in agreement to most of the following authors, pressure and enthalpy flow and temperature and a heat flux, for flow and heat transfer respectively.

3.2. Library Structure for Solar Thermal Power Plants

The models of the here presented library should be formulated as lumped to one dimensional models. Whenever possible, first principle equations should be applied. The models should allow to model plants on a decision maker to an early project developer level. The developed structure is presented below. It fits in the Stellenbosch University Solar Thermal Electricity (SUN*STEL*) project from which the name is derived.

1. SUNSTEL Library
- **1.1. Solar_Ambient**
- **1.2. Solar_Ambient_Package**
  - **1.2.1. Internals**
  - **1.2.2. Examples**
- **1.3. Irradiation_Transformer**
  - **1.3.1. Internals**
  - **1.3.2. Central_Receiver**
  - **1.3.3. Dish**
  - **1.3.4. Linear_Fresnel**
  - **1.3.5. Parabolic_Through**
  - **1.3.6. Examples**
- **1.4. Enthalpy_Transformer**
  - **1.4.1. Internals**
  - **1.4.2. Models (i.e. Compressor, Pumps, Turbine, Pipes, Combustor, …)**
  - **1.4.3. Examples**
- **1.5. Heat_Storage**
  - **1.5.1. Internals**
  - **1.5.2. Sensible**
  - **1.5.3. Latent**
  - **1.5.4. Chemical**
  - **1.5.5. Examples**
- **1.6. Control**
- **1.7. Internal**
- **1.8. Examples**
  - **1.8.1. ZolarOne_plant**

The structuring approach used in this work follows the path of the energy through the system i.e. the transformation of direct solar irradiation to power (Eck et al., 2010). Secondly the possibility to structure the system in subsystems is taken into account.

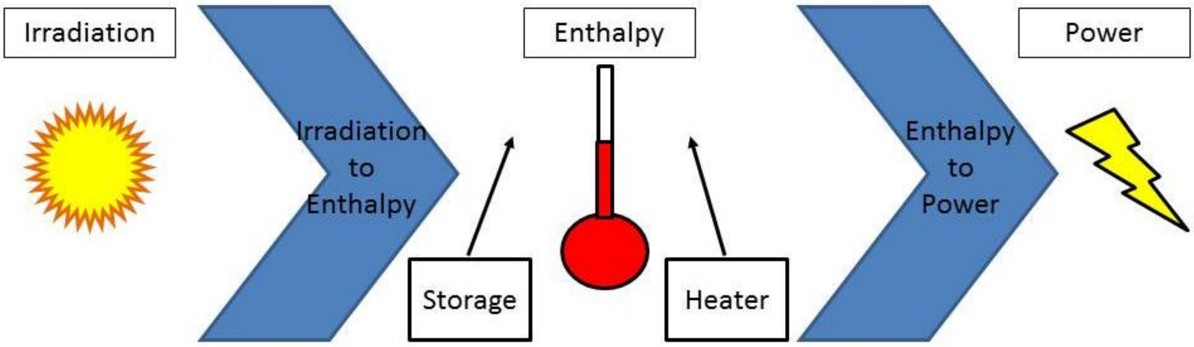

Figure 3-1 - Basic Structuring Approach – Tracking the Energy Conversion Through the System

The library is structured into five classes according to the five subsystems of a CSP plant. The classes are Solar_Ambient_Package, Irradiation_Transformer, Enthalpy_Transformer, Heat_Storage and Control. Though proposed by Eck et al., (2010) a transport subsystem is not taken into account.

All top level packages contain a subclass "Examples" where components out of this class can be tested independently and a subclass "Internal" where partial components and basic interfaces are stored which inexperienced users do not have to use (Richter, 2008). A colour convention according to Gräber et al. (Gräber, Kosowski, Richter, & Tegethoff, 2010) is established where ready-to-use components and libraries are marked green, models to build up component models are marked yellow, internal functions and partial models which inexperienced users do not necessarily have to access are grey (Gräber et al., 2010).

4.2. Inheritance structure and main concepts

Figure 3-2, shows the inheritance structure of an example model for a simple central receiver model. The partial models, PartialIrradiationToHeat and PartialIrradiationToEnthalpy are interface classes and are composed of the connector classes HeatPort and AmbPort and FluidPort and AmbPort, respectively. The connector classes HeatPort and Fluidport contain the necessary variables to exchange heat flow rates and enthalpy flow rates (Otter, 1999). The first principle equations such as a receiver energy balance are implemented in the Basic_Simple_Tower model, which inherits everything from its parent class PartialIrradiationToHeat. The ready-to-use component model Simple_Tower then accesses the Solar_Ambient component to retrieve the necessary solar inputs via the connector class AmbPort. A general flow model class ComplexPipe covers the flow of fluid through the control volume. This model can be derived depending on the simulation platform from other libraries or be self-written. All solar components can be derived using the two interface classes PartialIrradiationToHeat and PartialIrradiationToEnthalpy. Different levels of detail can be implemented therein.

Similarly the inheritance structure for modelling of a 1D latent heat storage device can be derived (Leonhardt & Müller, 2009). Considering Figure 3-3 the PCM_1D_cell model represents a control volume of a latent heat storage device. The interface

PartialThermalEnergyStorage provides the necessary interfaces to the Solar_Ambient component and a flow model, here again the ComplexPipe. The inheritance structure of the classes ComplexPipe, FlowModel, HeatPort, FluidPort, HeatCapacitor and HeatConductor are not shown in Figure 3-3.

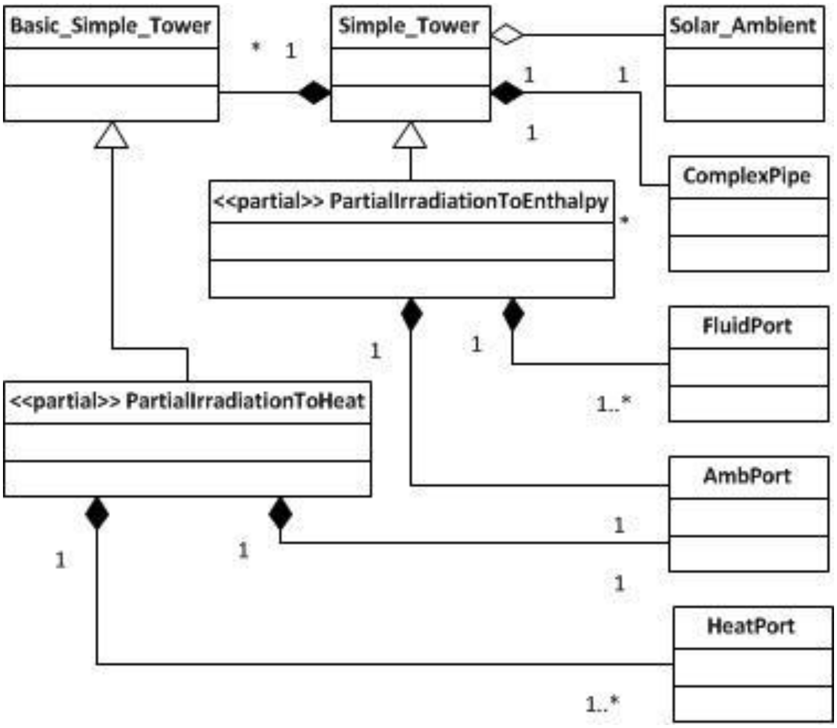

Figure 3-2 Class diagram of a Simple Tower Model

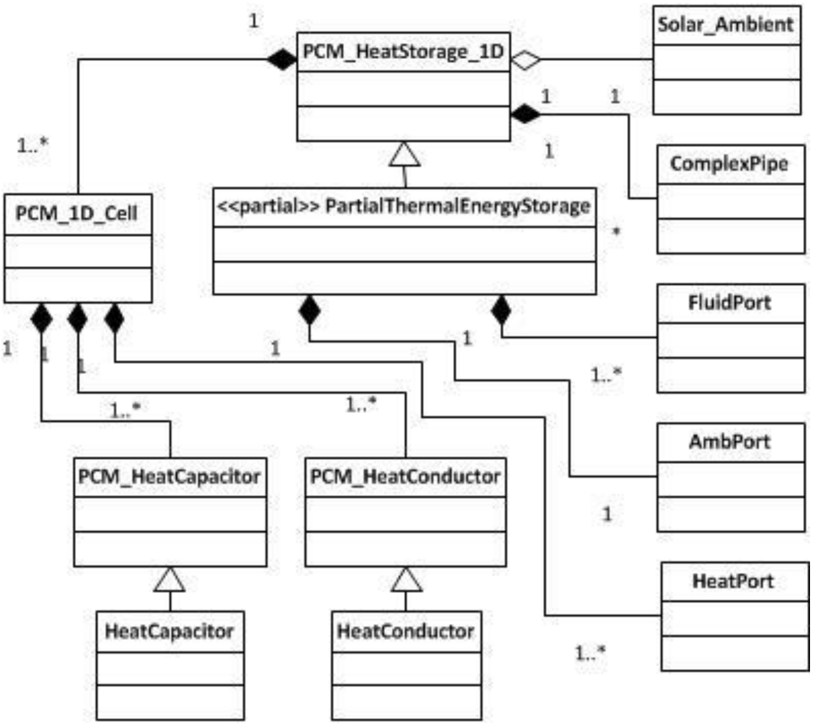

Figure 3-3 Class diagram of a 1D Latent Heat Storage Model

5. Solar_Ambient Component

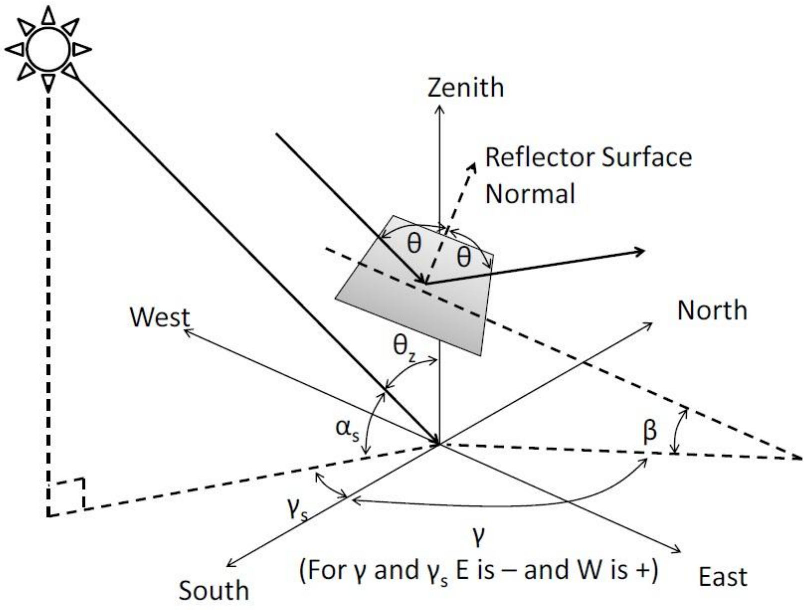

Figure 3-4 Horizontal coordinate system (Gauché, Pfenninger, Meyer, von Backström, & Brent, 2012)

The following equations are used to calculate the azimuth and elevation angle of the sun in a horizontal coordinate system (As seen in Figure 3-4).

The declination, δ, is calculated using the following equation (Cooper, 1969).

$$\boldsymbol{\delta = 23.45 \times \sin\left(\frac{360\ deg}{365\ d} \times (284 + N)\right)}$$ **Equation 3-1**

The unit of declination angle δ is degrees and the unit of Day N is days. The minimum and maximum value of the declination is -23.45 degree and 23.45 degree, respectively. Conventions are defined for a southern hemisphere winter, where the declination is bigger than zero and for a northern hemisphere winter declination is smaller than zero.

The apparent solar time T_{sol} is derived by the following equations (Duffie & Beckman, 2006; Iqbal, 1983; Spencer, 1972).

$$\boldsymbol{t_{sol} = t_{loc} + 4\ \frac{min}{deg} \times (\lambda_{loc} - \lambda_{std}) + E}$$ **Equation 3-2**

$$\boldsymbol{E = 9.87 \times \sin(2B) - 7.53 \times \cos(B) - 1.5 \times \sin(B)\,, E\ in\ [min]}$$ **Equation 3-3**

$$\boldsymbol{B = \left((N - 81) \times 0.989\right), \qquad B\ in\ [deg]}$$ **Equation 3-4**

t_{loc} is the legal (watch) time in hours, $\boldsymbol{\lambda_{loc}}$ is the local longitude in degrees, $\boldsymbol{\lambda_{std}}$ is the standard longitude of the corresponding time zone in degrees, E is the rotation speed correction in minutes. Unit of B is degrees, Unit of Day of the year $\boldsymbol{N}$ is in days.

The hour angle of the sun is calculated using the following equation (Duffie & Beckman, 2006).

$$\boldsymbol{\omega = \frac{360}{24}\frac{deg}{h} * (12 - t_{sol})}$$ **Equation 3-5**

The hour angle ω is in degrees t_{sol} is in hours. The following definitions are made: At zenith the hour angle equals zero, in the morning the value is bigger than zero, in the afternoon the hour angle is smaller than zero.

The elevation angle of the sun is calculated using the following equation (Duffie & Beckman, 2006).

$$\boldsymbol{\gamma_s = arcsin\{sin(\varphi) * sin(\delta) + sin(\varphi) * sin(\delta) * sin(\omega)\}}$$ **Equation 3-6**

γ_s is the elevation angle in degrees, φ is the latitude of the plant position in degrees, δ is the declination of the sun in degrees. ω is the hour angle in degrees.

The azimuth angle α_s of the sun is calculated using the following equation (Duffie & Beckman, 2006).

$$\boldsymbol{\alpha_s = arccos\left\{\frac{[sin(\varphi)*sin(\gamma)-sin(\delta)]}{[cos(\varphi)*cos(\gamma)]}\right\}}$$ **Equation 3-7**

φ is the latitude of the plant location in degrees, δ is the declination in degrees, γ is the elevation angle of the sun in degrees and ω is the hour angle in degrees. Note that the value is zero to the south, positive towards east and negative towards west.

4. Calculating the Sun Position in Flownex

Flownex SE is a thermo-fluid system simulation code developed by the South African company MTech Industrial. It is a flowsheet based environment which provides an elaborate component model library and Drag&Drop functionality to easily compose system models of flow- and heat-networks (see Figure 4-1). The implementation of models based on first principle equations is possible.

As a first attempt the Solar_Ambient component was implemented as a script in Flownex. The script allows other components and models to access hour angle, elevation angle, azimuth angle of the sun and solar time. Weather data and input parameters are stored in a Excel spreadsheet component.

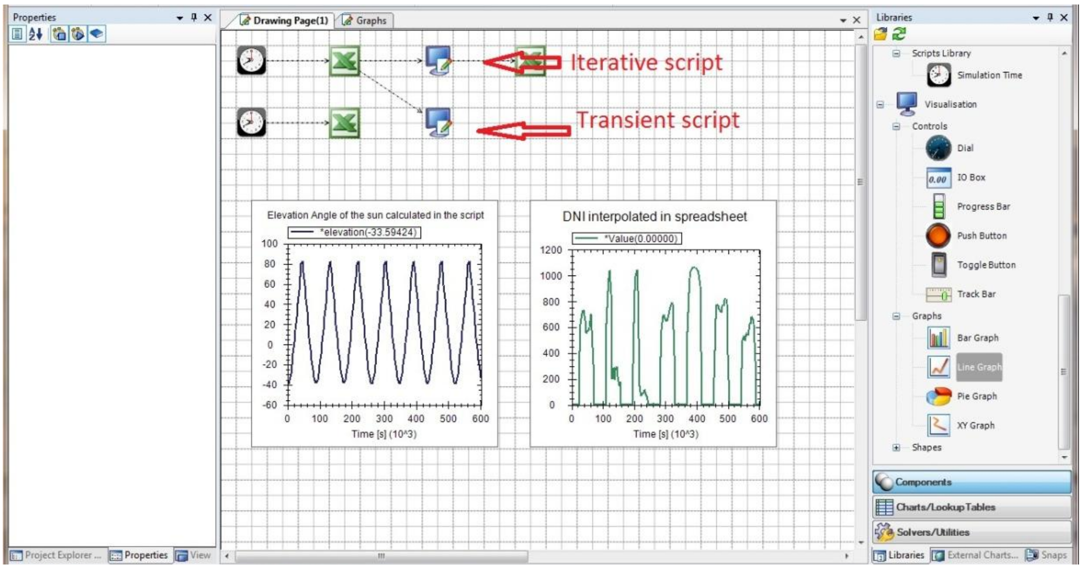

Figure 4-1 Screenshot of Flownex showing the necessary data links and calculation output

It needs as an input the current time step of the simulation, the local longitude, the standard longitude of the current time zone and the latitude. The experimental data can be copied in the spread sheet.

The time dependent data is read by using linear interpolation, interpolating the measured value depending on the current simulation time step.

Figure 4-2, p. - 10 - shows the DNI data for a week in Upington, Northern Cape, South Africa. Figure 4-3 and Figure 4-4 show the calculated azimuth and elevation angle of the sun. These two could be used for example in a 2D interpolation matrix which contains the field efficiency of a heliostat field.

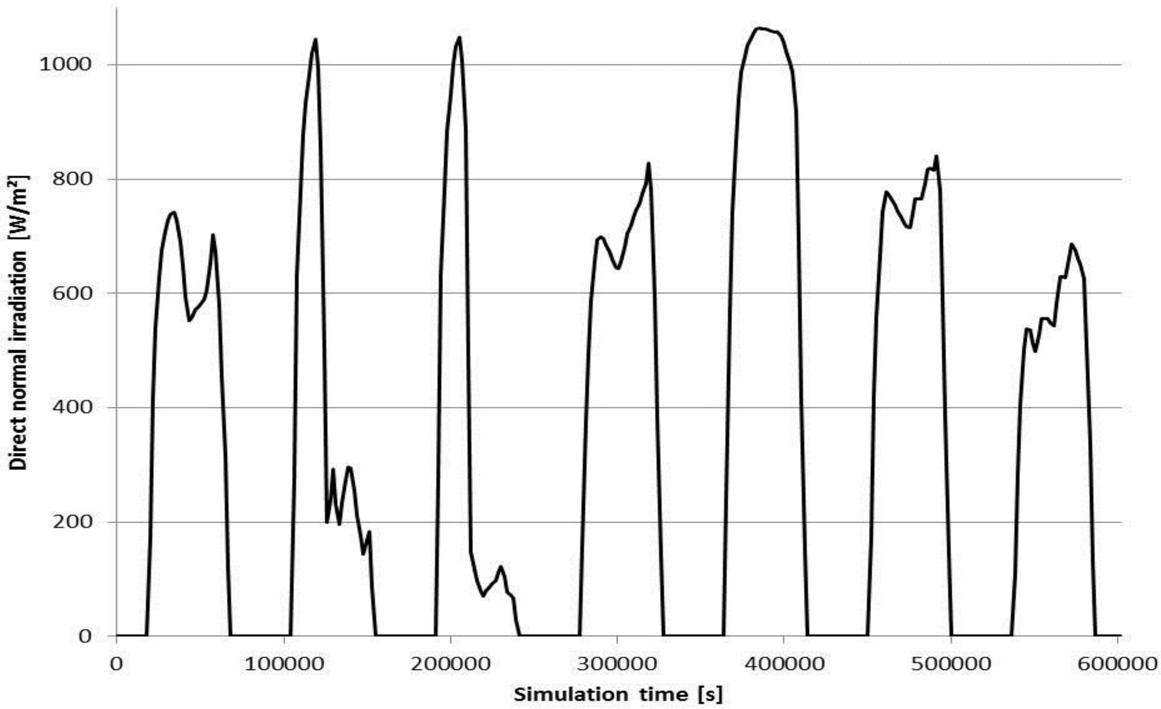

Figure 4-2 Direct Normal Irradiation (DNI) according to the weather data for the first week of the year in Upington, South Africa

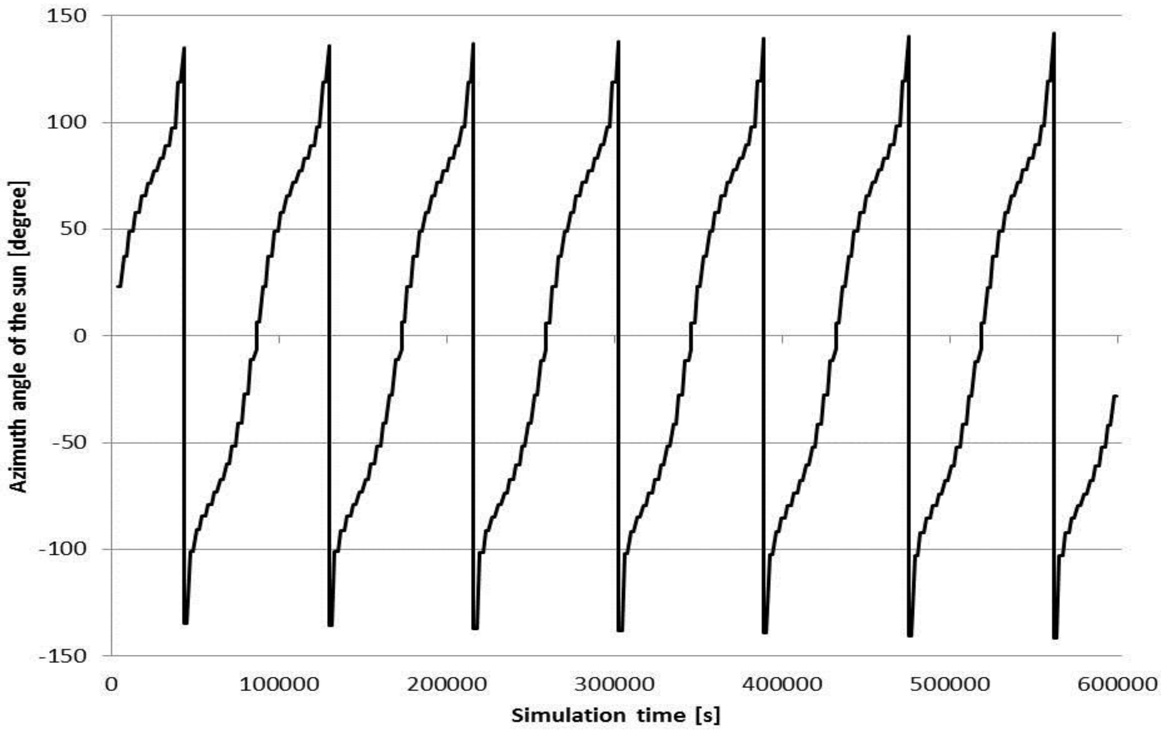

Figure 4-3 Azimuth angle of the sun for the first week of the year in Upington, South Africa. Steps occur due to hourly time steps of data

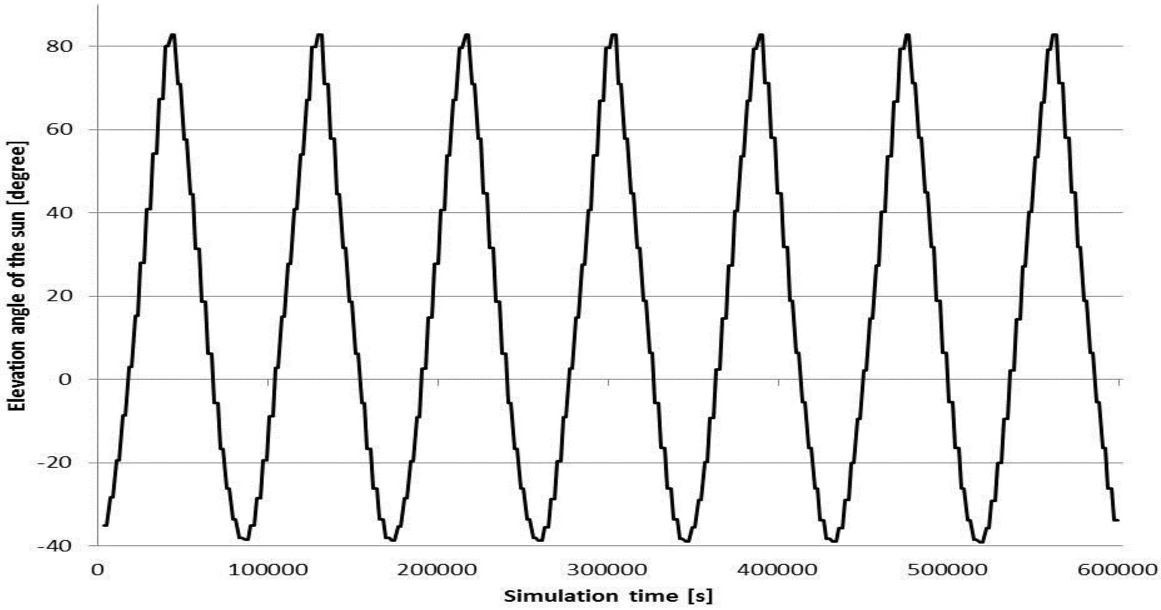

Figure 4-4 Elevation angle of the sun for the first week of the year in Upington, South Africa. Steps occur due to hourly time steps of data

5. Conclusion

In this paper we present a library structure for component models of solar thermal power plants using an object-oriented approach. We summarize existing library structures and give an overview of the related literature and methodologies. The necessary interfaces are defined and a main component which calculates the sun position and provides an interface to experimental data is implemented using the software environment Flownex

We give the minimum set of equations to determine the sun position in a horizontal coordinate system and specify the necessary input parameters.

Further research should assure that library and model development is aligned with the SolarPACES project “Standardization of CSP Performance Model Projection”, (STAMP). Full plant models including flow models for the Rankine and other cycles should be included.

As no standardized simulation platform is found until now, available codes should be carefully reviewed if they fit the needs.

6. Acknowledgments

I would like to thank all members of the Solar Thermal Energy Research Group for their support, advice and the opportunity to work on this project. At last I would like to thank my parents for their continuous support and belief.

7. References

Casella, F. (2003). Modelica open library for power plant simulation: design and experimental validation. *Proceedings of the 3rd International Modelica Conference*, 41-50.

Casella, F., & Leva, A. (2006). Modelling of thermo-hydraulic power generation processes using Modelica. *Mathematical and Computer Modelling of Dynamical Systems*, *12*(1), 19-33. doi:10.1080/13873950500071082

Cooper, P. I. (1969). Solar Energy. *Solar Energy*, *12*(3), 333-346. doi:10.1016/0038-092X(69)90047-4

Duffie, J. A., & Beckman, W. A. (2006). *Solar Engineering of Thermal Processes* (3rd ed.). John Wiley and Sons.

Eborn, J. (2001). *On Model Libraries for Thermo-hydraulic Applications*. Lund Institute of Technology.

Eck, M., Barroso, H., Blanco, M., Burgaleta, J.-I., Dersch, J., Garcia-Barberena, J., Gonzalez, L., et al. (2011). guiSmo : GUIDELINES FOR CSP PERFORMANCE MODELING – PRESENT STATUS OF THE SOLARPACES TASK-1 PROJECT. *Proceedings of 17th SolarPACES Symposium*. Granada, Spain.

Eck, M., Benitez, D., Hirsch, T., Ho, C., & Wagner, M. (2010). The first steps towards a standardized methodology for CSP electricity yield analysis. *Proceedings of 16th SolarPACES Symposium*. Perpignan, France.

Gauché, P., Pfenninger, S., Meyer, A. J., von Backström, T. W., & Brent, A. C. (2012). MODELING DISPATCHABILITY POTENTIAL OF CSP IN SOUTH AFRICA. *Proceedings of the 1th Southern African Solar Energy Conference* (Vol. 2011). 21-23 May, Stellenbosch, South Africa.

Gräber, M., Kosowski, K., Richter, C., & Tegethoff, W. (2010). Modelling of heat pumps with an object-oriented model library for thermodynamic systems. *Mathematical and Computer Modelling of Dynamical Systems*, *16*(3), 195-209. doi:10.1080/13873954.2010.506799

Hirsch, T., Eck, M., Buck, R., Dersch, J., Feldhoff, J. F., Giuliano, S., Hennecke, K., et al. (2010). Modelling, Simulation and Assessment of Solar Thermal Power Plants A First Step Towards Definition of Best Practice Approaches. *Proceedings of the ASME 2010 4th International Conference on Energy Sustainability ES2010* (pp. 1-9). May 17-22, Phoenix, Arizona, USA.

Iqbal, M. (1983). *An Introduction to Solar Radiation.* Toronto: Academic.

Leonhardt, C., & Müller, D. (2009). Modelling of Residential Heating Systems using a Phase Change Material Storage System. *Proceedings of the 7th Modelica Conference* (pp. 20-22). Como, Italy, Sep. 20-22. doi:10.3384/ecp09430025

Mühlthaler, G. (2001). *Anwendung objektorientierter Simulationssprachen zur Modellierung von Kraftwerkskomponenten*. Düsseldorf: VDI-Verlag, VDI Fortschritts-Berichte, Nr. 450, Reihe 6,.

Otter, M. (1999). Objektorientierte Modellierung Physikalischer Systeme. *at - Automatisierungstechnik*, *47*(1), A1-A68.

Richter, C. C. (2008). *Proposal of New Object-Oriented Equation-Based Model Libraries for Thermodynamic Systems*. TU Braunschweig.

Schwarzbözl, P. (2006). *A TRNSYS Model Library for Solar Thermal Electric Components (STEC) Reference Manual 3.0*.

Spencer, J. W. (1972). Fourier Series Representation of the Position of the Sun. *Search*, *2*(5), 172.

Tummescheit, H. (2002). *Design and Implementation of Object-Oriented Model Libraries using Modelica*. Lund Institute of Technology.