CHEMICAL ENGINEERING TRANSACTIONS

VOL. 39, 2014

A publication of
AIDIC
The Italian Association
of Chemical Engineering
www.aidic.it/cet

Guest Editors: Petar Sabev Varbanov, Jiří Jaromír Klemeš, Peng Yen Liew, Jun Yow Yong
Copyright © 2014, AIDIC Servizi S.r.l.,
ISBN 978-88-95608-30-3; **ISSN** 2283-9216

DOI: 10.3303/CET1439225

Software Tool for Creating and Modifying Parametric Shell and Tube Heat Exchanger Geometry

Tomáš Létal

Institute of Process and Environmental Engineering UPEI-VUT, Faculty of Mechanical Engineering, Brno University of Technology, Technická 2, 616 69 Brno, Czech Republic
letal@upei.fme.vutbr.cz

This paper presents software for creation and manipulation of parametric shell and tube heat exchanger geometry. Presented software is a key part in developing software for mechanical design or check of shell and tube heat exchangers according to (EN 13445 Unfired pressure vessels, 2009), which will also have optimisation capabilities. Presented software is using only freely available open source software.

A shell and tube heat exchanger can be described by a group of parameters. Structural analysis methods evaluate individual components separately, but they usually require (among other things) thorough knowledge of surrounding components geometries. In case of checking an already manufactured exchanger, a user has to input all the necessary parameters into software for structural analysis just once, and evaluate the design very efficiently. However, in case of design, where the final geometry is not known yet, a designer has to check multiple designs while adjusting some parameters and keeping track of their dependencies. The last task could be simplified if the exchanger geometry was modelled as a parametric assembly with automatic update procedure for changing dependent parameters.

Shell and tube heat exchanger geometry can be easily represented as an assembly of parts, such as shells, heads, flanges, tubes, tubesheets, baffles, saddle supports, etc. Most of these part classes have a few shape types which are not developed further. Such parts can be prepared in a database as parametric models. Afterwards, the model of exchanger geometry can be easily assembled using these parametric parts and constraints.

Currently there is no software which would be able to optimise mechanical design of shell and tube heat exchangers according to (EN 13445 Unfired pressure vessels, 2009). Since in general, the strength of a component depends on surrounding components, single components cannot be optimised separately. There is obvious need for parametric model tailored for purposes of heat exchanger mechanical design optimisation. Parametric geometric assembly is a key part of the model.

Presented software used history-based multi-level assembly design. Parts and subassemblies are absolutely positioned in local subassembly coordinate systems. Items in assembly tree are ordered and their defining parameters may depend on the parameters of previous items. Parameters and low level connections can be visualised as an acyclic directed graph.

Software was created in the Python programming language. Libraries PythonOCC and wxPython were used for creating geometry and graphical user interface respectively. A user can add, modify or delete parts of predefined types and high level connections between them. When modifying a part, user is allowed to directly change only independent parameters. Thanks to PythonOCC, created geometry can be easily exported to standard STEP format and therefore can be used in other software e.g. for FEM or CFD analyses.

1. Introduction

Shell and tube heat exchangers are usually designed iteratively. Design procedure starts with thermo-hydraulic design which is used to estimate main dimensions. Remaining dimensions have to be estimated with the help of structural analysis. It is usually more feasible to use standardized parts instead of ideal parts from the structural and thermo-hydraulic analyses perspective. Geometry in general plays an

Please cite this article as: Létal T., 2014, Software tool for creating and modifying parametric shell and tube heat exchanger geometry, Chemical Engineering Transactions, 39, 1345-1350 DOI:10.3303/CET1439225

important role in the whole design process as it presents necessary input for design analyses and it is usually changed and analyzed several times again before a final design is produced.

1.1 Current issues of shell and tube heat exchanger design optimisation

Shell and tube heat exchangers are used for high pressure applications and from the structural analysis perspective they are pressure vessels. Papers about structural optimisation of pressure vessels in general, such as (Carbonari et al., 2011) or (Choi et al., 2008), mostly deal with small subset of a pressure vessel. However, since neighbouring parts influence each other, it is hardly ever possible to optimise parts of the vessel separately with resulting optimal design of the whole vessel. The problem is extensive rather than complicated.

Using FEM analysis, it is unreliable to model a pressure vessel with solid elements because acceptable quality mesh would be computationally demanding. Instead, shell elements are often used. A shell model of a pressure vessel might be reliable for optimisation, but results of the stress analyses are evaluated using stress categories method, which has not been fully algorithmised yet.

Computationally most feasible option is using codes such as (EN 13445 Unfired pressure vessels, 2009). For such optimisation, parametric assembly model is required.

1.2 Specialized assembly modelling for shell and tube heat exchangers

Professional general purpose CAD modelling software products are normally used for parametric assembly modelling, and structural optimisation software could be built in as their extension. There are also specialized structural assessment software products for pressure vessels, which use simplified assembly models. The models are intended mainly for visual check of the design and the software products do not have optimisation capabilities.

2. Computer aided assembly design

Mechanical assemblies can be viewed as graphs, where the nodes representing sub-assemblies or parts and the edges representing constraints (spatial relationships between parts). With respect to the set of applied constraints, assembly can be either under-constrained, properly constrained or over-constrained. When some parameters of the assembly change, software should be able to estimate new positions of individual parts.

Individual parts are traditionally modelled with history-based approaches. In history based-design methods, parameters of a new feature can depend on parameters of previously defined features.

2.1 Assembly design methods

There are two main assembly design approaches: top-down and bottom-up. Bottom-up approach is typical for most of the CAD systems. Designers create individual parts and then they add them into assemblies as rigid parts using constraints. Bottom-up approach is limited, because the parts can be constrained just in a tree structure, but it is sufficient for wide range of assemblies.

Top-down approach is more traditional, but its implementation in CAD system is more difficult. It starts with assembly as a whole (skeleton assembly), which can be represented by a sketch. Individual parts and subassemblies are constrained to sketch key points and lines. According to Chen et al. (2012), top-down assembly design is still only partially supported by some CAD systems.

None of these approaches is supreme to another. Bottom-up approach is usually more convenient for smaller less complicated assemblies.

2.2 Constraints in assemblies

At the lowest level, constraints in general can be specified by nonlinear equations. To estimate new assembly layout, it is necessary to solve a system of nonlinear equations. For such tasks, CAD software use specialized solvers, tailored for specific types of equations, such as the one described by van der Meiden and Bronsvoort (2010). Such constraints are very flexible and intuitive to use.

Because connecting features of parts usually belong to a relatively small set of widely used connection types such as pin and hole or boss and groove, there have been many attempts to simplify constraints using this fact. Anantha et al. (1996) used assembly features – couples of entities on different parts, by which the parts can be joined together. Couples of assembly features can be joined together, constraining predefined degrees of freedom (DOF). Similar approach is used by Chang and Perng (1997). Slightly different approach was used by Singh and Bettig (2004), who describe “ports”, which should be defined on parts in their modelling phase and can be later used for automatic assembly procedure.

Update methods for estimating new position of parts lead to datum flow chain (DFC). DFC is an acyclic directed graph representing parts and their constraints and it is used to define sequence in which the parts placements are resolved.

3. Shell and tube heat exchangers assemblies

From the perspective of geometry, shell and tube heat exchangers are very specific engineering products. Main parts such as shells, ends, flanges, etc. have simple shapes which are most suitable for pressure loads and can be easily manufactured. There are papers on pressure vessel part shape optimisations, such as (Carbonari et al., 2011) or (Liu et al., 2001), but their results are usually not feasible using current manufacturing methods.

The heat exchanger assemblies can be decomposed to 3 main subassembly classes according to TEMA (2007):

- Front end stationary heads
- Shells
- Rear ends

These subassemblies can be decomposed further up to basic parts. Therefore, multi-level assembly model would be beneficial for describing heat exchanger assemblies. In the next chapter, multi-level aspects of shell and tube heat exchanger assemblies will be discussed in more detail.

3.1 Shell and tube heat exchanger constraints

Except for sliding contact of floating head, all constraints are rigid. Most of heat exchanger parts are either welded or bolted together. Special methods are used for connecting tubes to tubesheets, but the result is always a rigid constraint. If a part is fixed in an assembly model and all constraints are rigid, properly constrained assembly has to have a tree structure.

From the assembly perspective, shell and tube heat exchangers cannot be in general described just by tree structure. For illustration, assembly on Figure 1 shows shell and tube heat exchanger configuration with straight tubes and fixed tubesheets, which includes constraint-part loop: shell – tubesheet – tube bundle – tubesheet – shell.

For the purpose of part placement, it is sufficient to use rigid constraints in a tree structure. However, this is not sufficient for design check according to (EN 13445 Unfired pressure vessels, 2009), where all parts and rigid joint between them have to meet certain criteria. Therefore, assembly model should be able to describe all constraints, including over-constraining ones.

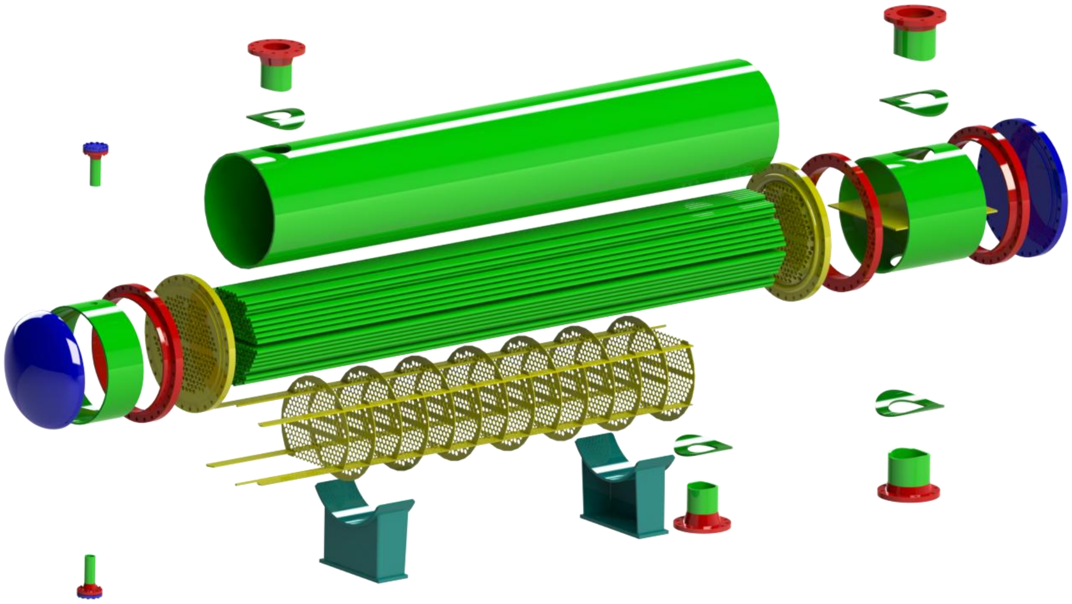


Figure 1: Exploded view of a shell and tube heat exchanger assembly

4. Multi-level history-based assembly design for shell and tube heat exchangers

This section presents a convenient assembly model for shell and tube heat exchangers, which can be applied to pressure vessels in general. The most important attribute of the software for the intended purpose of optimisation, is the ability to quickly update the assembly, when some parameters with dependencies change.

4.1 Multi-level aspects

In previous chapter it was concluded that shell and tube heat exchangers assemblies cannot be fully represented by a tree structure. However, a tree is the best structure for multilevel assembly models and it is still convenient for representing most of the assembly structure in general.

The presented software uses a multi-level assembly model with the tree structure where the root corresponds to the main assembly and branches correspond to parts and subassemblies. Individual parts and subassemblies are defined in their parent's coordinate system and they have parametrically defined translation and orientation with respect to the coordinate system. It is assumed, that there will be only rigid constraints in the tree and resulting assembly will be properly constrained.

4.2 History aspects

Before implementation of assembly modelling in CAD systems, assemblies were modelled as multi-body parts using history-based methods. Despite not being very flexible, this approach is still suitable for small rigid assemblies and it is relatively easy to implement.

As the main features of the ordered history list, presented software uses parts and subassemblies which are defined by sets of parameters. Parameters of a feature can be defined by an expression using the parameters of previous features from the list. The expressions act as additional constraints and help to capture design intent in a low level top-down fashion. If a parameter of a feature is changed, the feature and the features that are situated behind need to be updated as well.

Update method in history-based design updates parts in certain sequence which corresponds to history list. The parts with their dependencies can be described in general by acyclic directed graph, which corresponds to DFC.

4.3 Combination

A multi-level history-based assembly model can be easily used for describing assemblies with a graph structure of parts and constraints. Therefore, it is applicable for shell and tube heat exchangers as well as for many other pressure vessels.

5. Implementation

Presented software was implemented in the Python programming language. It is a high-level dynamically typed programming language. Python was chosen for its simplicity, which makes it very suitable for rapid software prototyping. It has also large set of freely available open source libraries.

The software was built exclusively using freely available open source software. It would be very difficult to make the software from the scratch. Using freely available libraries is naturally tempting, but it might not always be the best option, because their professional alternatives are usually more mature, stable and provide greater functionality. Considering simplicity of the implemented model, these libraries were more than sufficient.

Presented software uses PythonOCC for geometry visualization and export and wxPython for graphic user interface. PythonOCC is a wrapper for OpenCASCADE Technology (OCCT) library, which is platform for CAD/CAM/CAE software development. OCCT is used mainly for in house software development in large companies.

5.1 Assembly modelling library

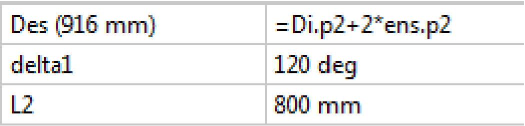
Core of the software is a library for modelling parametric heat exchanger assemblies. It has a tree-like data structure in which the root is the main assembly and the nodes correspond to parts and subassemblies. Each node has a unique identification, by which it can be easily accessed if needed. Identification is currently in format of letter "p" with number. Node identifications are displayed in GUI as a part of node captions, as can be seen in Figure 3.

Parts and subassemblies have local coordinate systems associated with the. Each part or subassembly is defined by its placement (rotation and position) in the coordinate system of its parent (sub)assembly and by geometry parameters. There are also additional parameters, such as materials, which are defined for future integration of structural assessment methods.

Each parameter can be defined either by fixed value or by an expression. The expression can be any valid math expression, which can use parameters of previous nodes from the assembly tree.

The expressions are stored partially as a text, except for referenced parameter names, which are stored as pointers to those parameters. In case where node identification would change, the expression would be derived with proper identification.

Figure 2 shows setting of three parameters – first one (Des) is defined by an expression which uses parameter name and its node, separated by dot. Current value of the expression is displayed after parameter label.



Des (916 mm)	=Di.p2+2*ens.p2
delta1	120 deg
L2	800 mm

Figure 2: Two possibilities of setting parameters in the GUI: by expressions or directly by values

Some part classes are predefined in a module which can be programmatically extended. The module currently includes shells (cylindrical and conical), ends (spherical and torispherical), neck flange, straight tube bundle and by one type of tubesheet, baffle and saddle support.

5.2 Graphic user interface

Initial motivation for graphic user interface was testing of developing core library, because testing using only scripts or command line would be tedious work. Currently, it is partially integrated in developing pressure vessel structural assessment software which is built on advanced user interface (AUI) module from wxPython library. This module allows definition of perspectives for different tasks such as assembly definition. The layout of assembly perspective is composed of 3 main custom widgets; a settings pane, PyCrust shell pane and a geometry viewer content window. The settings pane and the viewer content window are presented on the Figure 3.

The settings pane has 2 tabs; assembly tree view and component settings and it allows selecting and editing nodes from the assembly tree. The assembly is displayed in the geometry viewer content window and can be easily updated from the settings pane.

The Figure 3 shows assembly modelling perspective of the software with a shell and tube heat exchanger assembly. The assembly tree in settings pane is historically ordered from the top to the bottom. There is an apparent tree structure of a multi-level assembly.

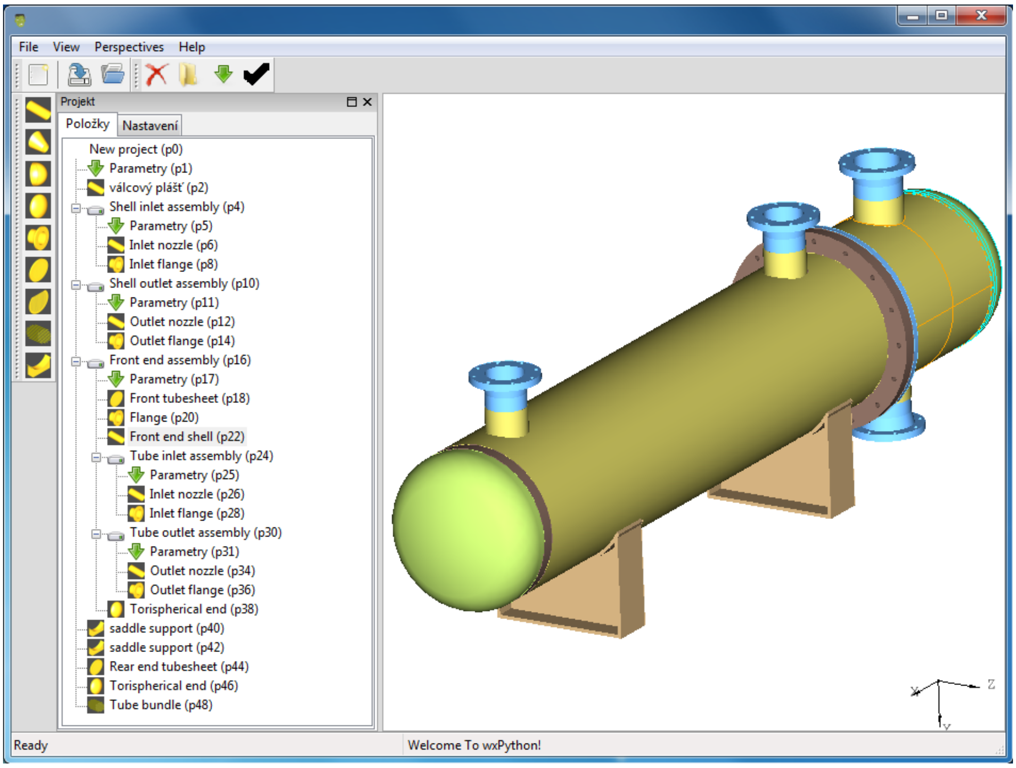


Figure 3: Shell and tube heat exchanger assembly created in the presented software

6. Future work

Connections between parts are defined just by low level expressions. Sets of such expressions will be predefined to simplify connecting part couples, e.g. flange joint or a tube bundle to a tubesheet. Such high-level connections will speed up assembly modelling.

The process of creating the geometry will be further simplified by predefining subassemblies according to TEMA (2007), which will be used for fast geometry creation.

The parametric geometry will be later used for automatised structural integrity assessment. In a future, such model could be used for multidisciplinary optimisation of shell and tube heat exchangers.

7. Conclusions

Presented paper deals with issues of shell and tube heat exchanger geometry modelling specifically tailored for mechanical design optimisation according to (EN 13445 Unfired pressure vessels, 2009). This standard provides methods for mechanical design check of separate parts in context of their neighbouring parts. Obvious optimisation algorithm could evaluate several design configurations, changing important parameters between evaluations and then choosing the best configuration based on some criteria. Naturally, parametric model of the whole heat exchanger assembly would significantly simplify this process. General assembly modelling methods are unnecessarily sophisticated and slow for this task.

The solution in form of the software for parametric modelling of shell and tube heat exchangers geometry was presented. Implemented multi-level history-based assembly model is simpler than currently used general assembly models and despite of being less flexible, it is still sufficient for fully constraining shell and tube heat exchanger assemblies using rigid constraints.

Acknowledgement

The results of this project NETME Centre Plus (LO1202) were co-funded by the Ministry of Education, Youth and Sports within the support programme „National Sustainability Programme I“.

References

Anantha R., Kramer G.A., Crawford R.H., 1996, Assembly modelling by geometric constraint satisfaction, Computer-Aided Design, 28, 707–722.

EN 13445 Unfired pressure vessels, 2009, Standardization Office of Mechanical Engineering and Rubber Industries, Courbevoie, France.

Carbonari R.C., Muñoz-Rojas P.A., Andrade E.Q., Paulino G.H., Nishimoto K., Silva E.C.N., 2011. Design of pressure vessels using shape optimization: An integrated approach, International Journal of Pressure Vessels and Piping, 88, 198–212.

Chang C.-F., Perng D.-B., 1997, Assembly-part automatic positioning using high-level entities of mating features. Computer Integrated Manufacturing Systems, 10, 205–215.

Chen X., Gao S., Yang Y., Zhang S., 2012, Multi-level assembly model for top-down design of mechanical products, Computer-Aided Design, 44, 1033–1048.

Choi W.-S., Kim T.-W., Seo K.-S., 2008, Shape optimization of a perforated pressure vessel cover under linearized stress constraints, Nuclear Engineering and Design, 238, 2468–2472.

Liu J.-S., Parks G.T., Clarkson P.J., 2001, Shape optimisation of axisymmetric cylindrical nozzles in spherical pressure vessels subject to stress constraints, International Journal of Pressure Vessels and Piping, 78, 1–9.

Singh P., Bettig B., 2004, Port-Compatibility and Connectability Based Assembly Design, Journal of Computing and Information Science in Engineering, 4, 197.

Sitharam M., Oung J.-J., Zhou Y., Arbree A., 2006, Geometric constraints within feature hierarchies, Computer-Aided Design, 38, 22–38.

TEMA, 2007, Standards of the tubular exchanger manufacturers association (No. 9th ed), Tubular Exchanger Manufacturers Association, New York, USA.

Van der Meiden H.A., Bronsvoort W.F., 2010, A non-rigid cluster rewriting approach to solve systems of 3D geometric constraints, Computer-Aided Design, 42, 36–49.