

Industrie 4.0 with MS-Excel?

Armin Fricke*, Jan C. Schöneberger

Chemstations Europe GmbH, Lindencorso, Unter den Linden 21, 10117 Berlin, Germany
armin.fricke@chemstations.eu

Industrie 4.0 is a project in the high-tech strategy of the German government to promote the computerization of traditional industries such as manufacturing. Other countries have similar programs.

Industrie 4.0 changes the cooperation of enterprises and the relationships of the company to its customers significantly. The company's success increasingly depends on the competence of the company to use software properly.

Not only the life cycles and the understanding of quality in the software industry and in process engineering come into conflict. Software ranges from the MS-Excel spreadsheet of the individual engineer over more or less closely associated applications of all sizes through to the group-wide integrating SAP R/3. At the interfaces of software, knowledge stocks of different domains (semantics) with their standardizations and with potentially different meanings collide.

In process design and development, chemical engineers use MS-Excel and process simulators to create process flow diagrams (PFDs), piping and instrumentation diagrams (P&IDs), equipment data sheets and heat and mass balances. Manufacturers typically use their own programs for the design of the equipment. Starting with the basic design or the following detailed design, asset management systems network and coordinate all the crafts involved in the planning and operation of a plant across sites and across the entire life cycle of the plant.

Various organizations, associations and initiatives such as Namur, Verein Deutscher Ingenieure (VDI), DEXPI, ISO 15926, CAPE-OPEN and others have begun to describe the requirements for data structures and the interchangeability of data records between CAE (Computer Aided Engineering) systems.

The paper shows in four examples from chemical engineering how various programs connect to a chemical process simulator through the existing interfaces. MS-Excel plays an important role in almost all cases. From these examples, course of action for process engineering, for computer science and for management derive.

1. Process Simulation

Process simulation on desktop computers dates back to 1981, when specialized packages became available, such as DESIGN II, ASPEN, SIMSCI (PROII), HYSYS, and CHEMCAD (Martín et al. 2014a).

Typical tasks that a process engineer carries out with a process simulator are flow sheeting, model development (e.g. fitting reaction kinetics from experimental data) and validation, equipment sizing (heat exchangers, vessels, piping, valves, pumps, columns, etc.), thermophysical property calculations (bubble/ dew points, data regression for pure components and binary interaction parameters, etc.), scale up from lab to pilot and pilot to production, dynamic simulations (control system design, safety analyses), process intensification studies, energy efficiency/optimization, data reconciliation, process economics and troubleshooting/ process improvement.

The term "process simulation" encompasses all these tasks. Although the commercial products mentioned above vary in the details of their functionality and their usability, they provide a similar support for the process engineer carrying out the tasks. All examples in this work have been implemented with CHEMCAD Version 6.5.6 (Chemstations, 2014).

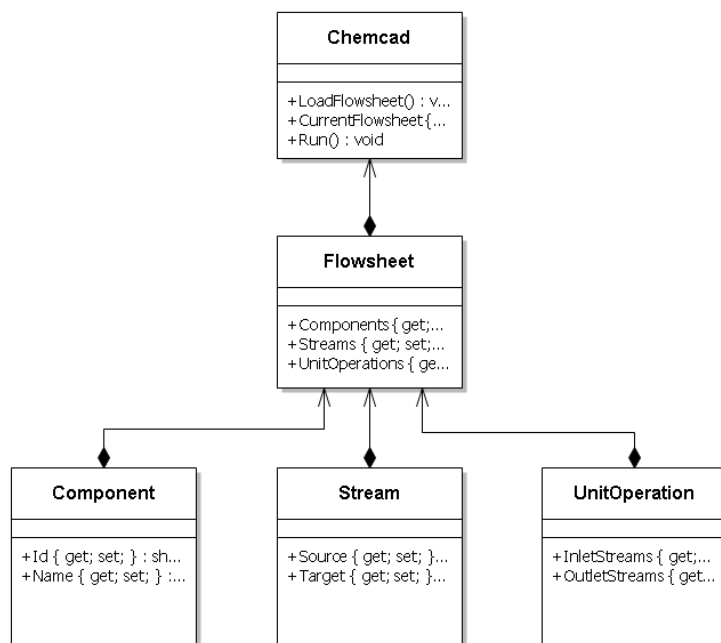


Figure 1: Object model of a process simulator

Modern software engineering follows a model-based approach where code is a derivation of an object model. Object-orientation is the logical extension of older techniques such as structured programming and abstract data types. An object has state (data) and behavior (code) with the addition of polymorphism and inheritance. As objects correspond to things found in the real world, an object model can be understood both from the domain expert and from the software engineer.

Figure 1 shows the object model of a process simulator. Starting from the top of the hierarchy, the first information object is the process simulator itself with the functions (behavior, code) to load and run the current flow sheet. A flow sheet consists of components, streams and unit operations. A stream has a source and a target unit operation whereas a unit operation has incoming and outgoing streams. Note that Figure 1 shows all objects of the top three levels of the object model but shows only a few examples for the data and the code of the objects.

The vendors of a process simulator continuously enhance its functionality. Drivers for the decision to implement new functionality are the quality of published scientific models and an anticipated market demand that warrants the investment for the implementation of new functionality into the process simulator. The economic return-on-investment is not the only criterion for the decision whether to implement new functionality since each new functionality makes the process simulator more complex and potentially less intuitive to use for the large majority of existing customers. Although the majority of the customers is satisfied with the available functionality, a minority requires very specific functionality that is not available in any of the commercial process simulators.

2. Process Simulators and MS-Excel

All process engineers due to its ease of use and availability probably use MS-Excel. Therefore, process simulators typically allow connecting to MS-Excel to enhance the functionality of the process simulator by customer specific simulation models and algorithms (Example No. 1):

- MS-Excel DataMap: Transfer any value from the flow sheet in the process simulator to a cell in a MS-Excel worksheet and vice versa, e.g. to add a simple cost function to the PFD or to export an equipment data sheet for the detailed engineering by an equipment manufacturer.
- Visual Basic for Applications (VBA) is an implementation of Microsoft's event-driven programming language Visual Basic 6 and is built into most Microsoft Office applications and in CHEMCAD. It is used to automate repetitive tasks or to build new capabilities into Excel, e.g. to develop new algorithms, to analyze data, and then to use the charting capabilities in Excel to display the results (Microsoft, 2009). Compared with MS-Excel, VBA as an imperative programming language is better suited to implement control flow

statements (e.g. continuation at a different statement, executing a set of statements only if some condition is met, executing a set of statements several times, until some condition is met, executing a set of distant statements, after which the flow of control returns, etc.) and to work with variables. VBA can be used to control one application from another via OLE Automation.

- Custom Mixing Rules: Mixing rules determine how the process simulator calculates the properties of a mixture, based on the properties the pure components. Custom mixing rules programmed in VBA can replace the built-in liquid density mixing rule or the built-in liquid viscosity mixing rule.
- Custom Kinetic Rate Reactions: A custom rate expression, defined in a MS-Excel spreadsheet, replaces the pre-defined generic kinetic rate equation.
- Custom Unit Operations: Custom unit operations allow specific equations to calculate heat and mass balances for e.g. membrane separation units, fuel cells, specialized solids handling units, etc. Custom Unit Operations can be implemented in MS-Excel, VBA, C++, FORTRAN or any other .NET programming language.

MS-Excel “is a perfect tool or interface not only to perform calculations but also to connect different software so that the end user can interact with Excel, and behind the scenes other software such as CHEMCAD®, MATLAB®, Mathcad®, or GAMS® is running and reporting the results back to Excel.” (Martín et al. 2014b). The process engineer working on the typical tasks mentioned in Section 1 is in charge of enhancing the functionality of the process simulator as mentioned above. Process simulator vendors offer support through their own resources or through external resources at an additional charge, which is not included in the software license fee.

3. Process Simulators and other Software Packages

Whereas Example No. 1 in the previous chapter describes several ways of enhancing the functionality of a process simulator, this chapter focusses on the links of the process simulator to products of other players in the engineering value chain (Porter 1985). Other value adding activities (Porter 1985) connected to process simulation are –in the basic design phase of process engineering– providing property data for a process simulator (Example No. 2) and embedding a process simulator in a material flow network (Example No. 3). In the detailed engineering phase, a process simulator can link to asset management systems (Example No. 4).

3.1 Property Data

Typically, process simulation programs provide property data from specialized vendors. “The Design Institute for Physical Properties (DIPPR) is the world’s best source of critically evaluated thermo-physical and environmental property data.” (DIPPR 2014). DIPPR Project 801’s purpose is to provide accurate and complete thermodynamic and physical property data for process engineering. The database contains data for 32 fixed-value properties and 15 temperature-dependent properties of 2047 chemicals and is included in CHEMCAD.

“Dortmund Data Bank is the worldwide largest factual data bank for thermophysical properties of pure components and their mixtures.” (DDB 2014) DDB-Connector for CHEMCAD is a standalone program, which provides an easy connection from CHEMCAD to the Dortmund Data Bank with three main functionalities:

- Properties of a component system in a simulation can easily be searched in Dortmund Data Bank and transferred to CHEMCAD.
- Binary VLE data calculated in CHEMCAD can be plotted and compared with experimental datasets in Dortmund Data Bank.
- A component pair can be selected from the simulation, and the plotting program of Dortmund Data Bank can be used to visualize the CHEMCAD calculated results.

DDB-Connector for CHEMCAD can also be used to import physical properties from any external data source, e.g. physical property estimation data from CRANIUM 2.0 (Molecular Knowledge Systems, 2014) into CHEMCAD via the so-called neutral file format. A prototype implementation of the DDB-Connector was programmed in VBA and the final software was programmed in the programming language C#.

3.2 Material Flow Analysis

In the material-intensive chemical industries, substantial increases in resource efficiency require an integrated analysis and optimization of complex production systems. The optimization must include the use of raw material and energy, the resulting costs and the environmental and climate impacts (Denz, 2014). Models for

this optimization combine industrial ecology and especially life cycle analysis with process simulation and heat integration (Denz, 2014).

Material flow networks (MFN) and flow sheet simulation represent two complementary modeling techniques, each showing strengths and weaknesses. MFN especially suit coarse-grained modeling of larger-scale processes and periods with a focus on the economic and ecological analysis of the used materials and energy. Flow sheet simulation focusses on rigorous physical modeling of chemical processes and takes into account the conservation equations and the laws of thermo-dynamics (Denz, 2014).

A pragmatic master/ slave architecture is based on transition refinement and data exchange via spreadsheets. The approach was prototypically implemented using the software tools Umberto, CHEMCAD, and MS-Excel and successfully applied to the example of a steam generator (Denz, 2014).

An example shows a coupled model of a steam generator: The main model is a material flow network in Umberto. Enthalpy flows are shown in the Sankey diagram. A flow sheet simulation in CHEMCAD details the transition that represents the boiler.

3.3 Asset Management

Asset management is a systematic business process of deploying, operating, maintaining, upgrading, and disposing of fixed assets –also known as "tangible assets" or property, plant, and equipment (PP&E)- cost-effectively. COMOS from Siemens AG is a software solution for an object-oriented and cross-departmental plant management over the entire plant lifecycle.

One of the most important tasks of the front-end engineering and design (FEED) phase is an initial cost calculation and a clear representation of processes in a PFD. All necessary data sheets and lists, such as equipment plans, material information, etc. can be generated automatically. COMOS FEED offers a library with modules that can be copied, moved and reused to standardize recurring sequences. An open interface model ensures flexibility and data consistency during import and export of data. The interfaces can be used to directly access the process data from different simulation programs. Data can be seamlessly imported and integrated into COMOS and can be used to create PFDs in COMOS FEED. The simulation objects are linked with PFD objects in COMOS. Once the PFD is complete, all data and information from the FEED phase is converted to P&IDs in the next step (Siemens 2014).

For demonstration purposes, an Ethylene Glycol Process (according to Melham 2004), which consists of four reactions, has been modeled in CHEMCAD using VBA for a custom reaction rate. The flow sheet is transferred into a COMOS PFD and subsequently into a COMOS P&ID via a programmed routine.

3.4 Conclusions from Examples 1 to 4

The examples have in common that MS-Excel plays a central role in extending the functionality of software programs or in connecting a process simulator to other software programs. Depending on the point of view, a software vendor sees this as an extension of the process simulator or as an extension of the other software program that connects to the process simulator. MS-Excel becomes the technical basis for an interface between software packages.

Software vendors regularly deliver updates for their programs to fix bugs and to deliver new functionality. Microsoft publishes updates on a monthly basis. Many of these updates concern security issues in the operating system and all the MS-Office programs including MS-Excel. Chemstations publishes updates for CHEMCAD every 3-4 months. Siemens publishes updates for COMOS quarterly. Most software vendors follow a regular schedule of several months.

The life cycle of a software package may be as long as the life cycle of a piece of equipment or an entire plant, i.e. in the range of 10-30 y and even above. However, software is updated regularly and bug fixing –which is called maintenance in the software industry- is common in cycles of 1-3 months. Changes to one software package may therefore influence the overall functionality of two linked software packages.

4. Interfacing Options

An interface allows reading from and writing to the objects of another software package. An interface exposes a selection of the functions of its objects in the form of a so-called class library. The class library for CHEMCAD is published on (CGC, 2014).

Therefore, the objects of the process simulator are re-used ("persist" as computer science calls it) beyond the boundaries of the process simulator program. Interfaces are at the same time a technical and a semantic bridge between software programs. Technical bridges share a common technical standard, whereas semantic bridges share a common knowledge about the domain. Depending on the purpose of the system, different

objects might be found on either side and therefore, objects of a process simulator and objects of adjacent programs will always differ in meaning. An important decision for the design of an interface lies in the standardization of the meaning (semantics) in the interface.

The technical bridge can be implemented with a file or data transfer or with function calls or with shared objects. The transfer option is the easiest to implement but has the disadvantage of duplicating information without a link that records the changes of each clone (creating independent “instances” as computer science calls it). The function call option is easy to implement but heavily dependent on the architecture and internal changes of the called program. The shared object option captures only those details about an object that are relevant to the interface (computer science calls this “abstraction”). It requires an object model (as described above) and can only be implemented inside a program – in this case the process simulator. A process simulator programmed in FORTRAN is unlikely to have an internal object model. Only few process simulators expose their object model and class library. MS-Excel can only use the transfer and the function call options and needs VBA to use the shared object option.

The technical bridge can be implemented “from inside” or “from outside” the process simulator. “From inside” describes simulator-internal code that accesses another software program’s objects via the other program’s published classes, e.g. as a custom unit operation described in chapter 2. “From outside” describes any software program and any programming language calling functions of the process simulator. As an example, a simple program in C# or Java could start CHEMCAD, open the flow sheet “MyFlowsheet.CC6”, set the temperature of stream No. 3, run the flow sheet, get the heat duty in unit No. 17, close the flow sheet and stop CHEMCAD. This example illustrates that choosing between “from inside” versus “from outside” is an architectural decision about which side “directs” the overall program logic. MS-Excel with or without VBA can play both roles.

The semantic bridge can have the form of a coupler, which standardizes task-task communications, controlling tasks and concurrent execution. Commenting on the success of industry standards like CAPE-OPEN, (Preisig, 2013) rightly states that “each party was protecting its own interests and the best compromise for all parties involved took the form of wrappers” which he defines as combinations of programming language with communication library codes. “Compared with couplers, a [pure programming] language approach avoids the passing of irrelevant information and leads to higher efficiency (Preisig, 2013).”

The semantic bridge can have the form of an Ontology. Ontologies provide a shared and common understanding of a domain and an explicit conceptualization that describes the semantics of the data. Munoz (2013) demonstrates that “the use of an ontological model for representing a waste treatment plant (...) provides more accurate information to the optimization tools which will lead to better solutions based on the plant constraints.”

MS-Excel with VBA cannot be integrated into a coupler and they are unable to use ontologies.

This section shows that extending a process simulator’s functionality with other software programs is a complex undertaking. Process engineers are usually not equipped to take the required decisions about software architecture or implementation alternatives, especially, if software life cycle issues must be considered.

5. Evaluation of MS-Excel

MS-Excel is widely accepted among engineers because it is available to most engineers (“free”), highly individual and easy to use like a pocket calculator. Summarizing the capabilities of MS-Excel including VBA, it extends a process simulator and can be used to design an interface to other software programs if a restriction on the technical bridge level is acceptable. This makes MS-Excel with VBA a good tool for an initial implementation of algorithms, methods or models to demonstrate a feasibility and for building prototypes.

MS-Excel including VBA fails when an interface must be “semantic-aware” since handling objects and connecting to ontologies is beyond its scope and functionality.

MS-Excel also fails when it comes to the deployment of the application, i.e. to all of the activities that make a software available for use to other people than its creator. There is no protection of intellectual property and no quality assurance for usability and documentation. The engineer has no tools that support versioning and branching of his work. MS-Excel was not designed to be a data management system that additionally documents the acquisition, sources and quality of data, the so-called metadata.

Language-oriented programming, e.g. in C# or Java for interfaces to a process simulator can solve both the non-“semantic-awareness” and the deployment issues. A pre-requisite is the availability of a class-library that interfaces the object model of the process simulator.

6. Conclusions

This work has presented technical and semantic interfaces between software for process simulation and other software packages commonly used in process engineering. It addresses a major issue that comes with market changes referred to as Industrie 4.0: How to use software effectively and efficiently in the chemical, pharmaceutical and life science industries.

Process engineering suffers from the demand to speed up the value creation process. The continuously rising use of software increases this effect more than it can reduce it. The tension between process engineering, which lives in 30-year life cycles of a production plant, and software engineering, which delivers 3-month updates is not likely to disappear.

Overall, the main contribution of this work allows process engineering, computer science and management to adjust their value creation to the market changes:

- Process engineering has to separate IT development from IT deployment.
- Computer science has to develop architectures that enable collaboration between all combinations of humans and machines/ devices (or so-called “things” in the Internet-of-Things). They must include “security by design”, provide interfaces to ontologies or must be able to handle large amounts of machine data (“big data”).
- Management (including IT management) has to define business processes and has to implement the IT systems that support the business processes. This includes the deployment of IT solutions for the process-engineering department.

It is good practice to start with a prototype. However, companies need stable software that can endure 30 years. A worksheet in MS-Excel might not fulfill this latter requirement.

References

- CGC Capital-Gain Consultants, 2014, <cc-api.com>, assessed 30.11.2014
- Chemstations, 2014, <chemstations.com> and <chemstations.eu>, assessed 30.11.2014.
- DDB 2014, Dortmund Data Bank Software & Separation Technology GmbH, www.ddbst.de, assessed 30.11.2014
- Denz N., Ausberg L., Bruns M., Viere T., 2014, Supporting Resource Efficiency in Chemical Industries - IT-based Integration of Flow Sheet Simulation and Material Flow Analysis, *Procedia CIRP*, 15, 537–542, doi:10.1016/j.procir.2014.06.060
- DIPPR 2014, Design Institute for Physical Properties, www.aiche.org/dippr/projects/801, assessed 30.11.2014
- Martín M. M., Grossmann I. E., 2014a, Modeling, Simulation, and Optimization in the Chemical Engineering Curriculum, in: Martín M. M., Ed., 2014, *Introduction to Software for Chemical Engineers*. CRC Press, London, United Kingdom.
- Martín M. M., de Juan L. M., 2014b, Excel for Chemical Engineering, in: Martín M. M., Ed., 2014, *Introduction to Software for Chemical Engineers*. CRC Press, London, United Kingdom.
- Melhem G.A., Gianetto A., Levin M.E., Fisher H.G., Chippett S., Singh S.K., Chipman P.I., 2004, Kinetics of the reactions of ethylene oxide with water and ethylene glycols, 2004, *Process Safety Progress*, 2001, 20(4), 231–246, DOI: 10.1002/prs.680200405
- Microsoft, 2009, Getting Started with VBA in Excel 2010, msdn.microsoft.com/en-us/library/office/ee814737(v=office.14).aspx, assessed 30.11.2014
- Molecular Knowledge Systems, 2014, <www.molecularknowledge.com>, assessed 30.11.2014.
- Munoz E., Capon E., Hungerbuhler K., Espuna A., Puigjaner L., 2013, Decision Making Support Based on a Process Engineering Ontology for Waste Treatment Plant Optimization, *Chemical Engineering Transactions*, 32, 277-282, DOI:10.3303/CET1332047
- Porter M.E. (1985), *Competitive Advantage: Creating and Sustaining Superior Performance*, New York.: Simon and Schuster, USA.
- Preisig H.A., Lobo S.C., 2013, The Role of Couplers in Chemical Model-based Engineering, *Chemical Engineering Transactions*, 32, 1405-1410, DOI:10.3303/CET1332235
- Siemens (2014), w3.siemens.com/mcms/plant-engineering-software/en/comos-process/comos-feed/Pages/Default.aspx, assessed 30.11.2014