Ian David Lockhart Bogle and Michael Fairweather (Editors), Proceedings of the 22nd European Symposium on Computer Aided Process Engineering, 17 - 20 June 2012, London.

Information integration: Generating functional models from structural ones

Manuel Rodríguez , José L. De la Mata, M. Eugenia Alvarez

Abstract

Information integration is a very important topic. Reusing the knowledge and having common representations have been (and it is) an active research topic in the process systems community. Conventional (structural) But only structural models have been dealt with so far. In this paper the issue of integration is related with two types of different knowledge, functional and structural. Functional representation and analysis have proved very useful, but still it is developed and presented in a completely isolated way from the classic structural description of the process. This paper presents an architecture to integrate both representations.

Keywords: Process representation, functional models, structural models.

1. Introduction

Information integration is a very important topic. Reusing the knowledge and having common representations have been (and it is) an active research topic in the process systems community. Some ontologies have been developed, as OntoCape Morbach et al. (2007) and some neutral model formulations have been presented (as CapeOpen) in order to promote this information sharing and integration. But these approaches have dealt only with structural models and with the problem of merging and using information from the different software applications used [Wiesner et al. (2011)]. This paper addresses the issue of integrating information between the structural and the functional views of a single system. The remaining of the paper is organised as follows, section 2 introduces de D-higraph functional modeling methodology, the third section presents the developed tool for the integrated use of structural and functional models, section four shows an application of the proposed architecture and finally section five presents the conclusions of this work.

2. D-higraph methodology

This section briefly introduces D-higraphs, their elements, properties, representation and application. For further information and deeper understanding of the methodology, the reader is encouraged to have a look at De la Mata & Rodríguez 2010b

2.1. D-higraph: Dualization of Higraphs

Higraphs are a general kind of diagramming objects well suited to the behavioral specification of complex concurrent systems. They were first presented by Harel (1987) and they can be considered as an extension and combination of conventional graphs and Venn diagrams. Higraphs consist of two elements, blobs (states) and edges (transitions) connecting the blobs. However, higraphs are not suitable for process systems specifications.

Rodríguez & Sanz (2009) first presented D-higraphs as a functional modeling technique that merges functional and structural information of the system modeled.
They came from the dualization of Higraphs: blobs represent transitions and edges represent states. Disjoint blobs imply and AND relation, i.e., both transitions between states take place. Orthogonal blobs represent and OR relation, i.e., only one of the transitions takes place.
It has to be noticed that a D-higraph is NOT a dual higraph (like dual graphs), obtained from changing blobs by edges and edges by blobs. The duality lies in the interpretation of blobs, edges and their properties.

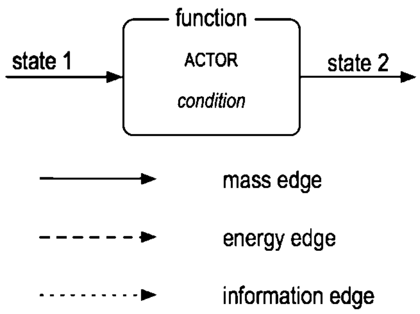

Figure 1. Basic blobs and types of edges

2.2. Main components

Blobs represent functions (transitions) and they are depicted with their elements as shown in the top of Fig. 1. The function is performed by the ACTOR producing state 2 if the state 1 is enabled and if the condition is true.
Firing the function causes new states, represented by edges coming out of the blob. Edges represent flows of mass, energy or information, which are responsible of all of the interactions in a process system (Lind, 1994).
Mass, energy and information edges are depicted differently, as shown in the bottom of Fig. 1. However, the type of flow does not affect the behavior of the model, it is a visual aid to represent more information.

2.3. Properties

• *Blob connection.* An edge always links two blobs: its tail and its head. Under certain conditions, the blob in the tail or in the head can not be represented (elliptic blob) but it exists.
• *Blob inclusion.* Blobs can be included inside of other blobs (Venn diagrams inclusion property). This means that the inner blob performs a function that is necessary for the function of the outer blob. This is how hierarchical functions are represented and how structural and functional information is integrated.
• *Partitioning blobs.* A blob can be partitioned into orthogonal components, establishing an OR condition between the partitions.
The main objective of D-higraphs is not only the representation of knowledge about process systems. De la Mata & Rodríguez (2010a,b) provide a series of causation rules relating two events (cause and effect) that allow us to track the evolution and propagation of failures across the system. This rules combined with sensor data of the process enables the possibility of performing FDI analysis using D-higraphs models.

3. Information integration tool

3.1. P&I representation

The available P&I representation of a process is translated to a steady state model, in this case to an Aspen Plus model. This initial translation is performed manually. The steady state model is converted to a dynamic model (Aspen Dynamics / Aspen Custom Modeler). This is the structural representation of the process.

3.2. D-h tool

The steady state model is automatically translated to D-higraphs model, additional information regarding the functionality of the different units is provided to the API to perform the creation of the model. The developed tool, Álvarez (2010), allows a visual representation of the goals of the process.

3.3. CLIPS

The D-higraph representation is automatically translated to CLIPS [CLIPS (2011)] which is a production system (rule based) that will be used to conduct fault diagnosis and HAZOP analysis.

3.4. Architecture

Figure 2 shows all the components assembled. The communication between the structural (dynamic) model and the functional one is through an extension coded as a dll. This dll allows controlling the simulation as well as being an input to the functional representation of some faults in order to analyze the effects on the whole process.

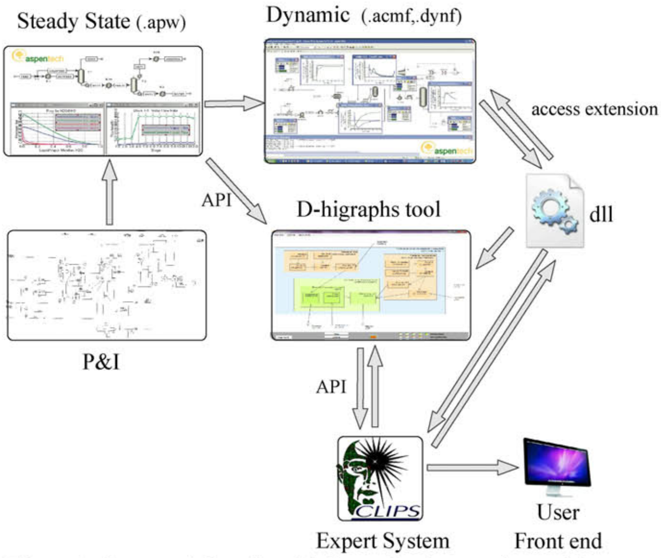

Figure 2. Structural-functional information integration architecture

There are different uses of this environment from conventional simulation to fault diagnosis, sensor validation, control reconfiguration or HAZOP analysis.

4. Application

The process used consists of a jacketed reactor, a cooler and a storage tank. An exothermic reaction of isomerization in liquid phase is carried out in the reactor. Cooling water is poured through the jacket to keep reactor temperature. The cooler is a shell and tube heat exchanger with one shell pass and two tubes passes. There are also five control loops controlling the reactor level and temperature, cooler outlet temperature, tank level and recycle flow. The P&ID of the process is shown in Fig 3

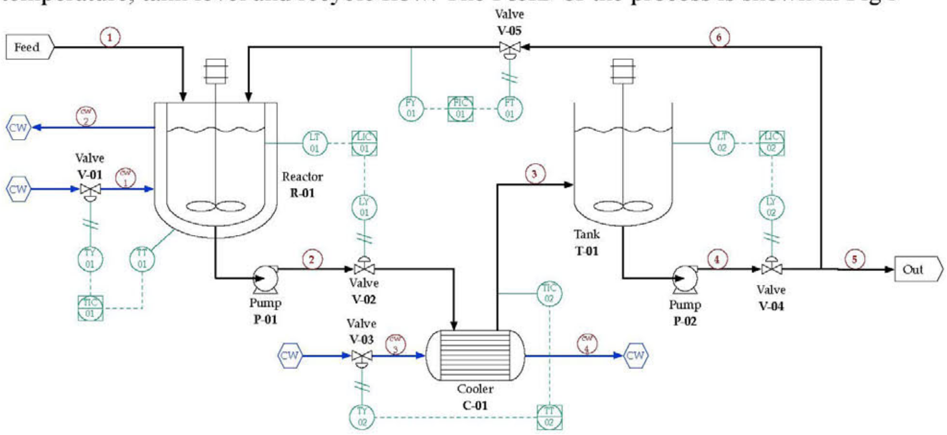

Figure 3. Piping and instrument diagram of the test process

For this P&I an steady state model as well as a dynamic model has been created. A functional model has also been generated using the D-higraph tool, figure 4 shows the functional model of the reactor.

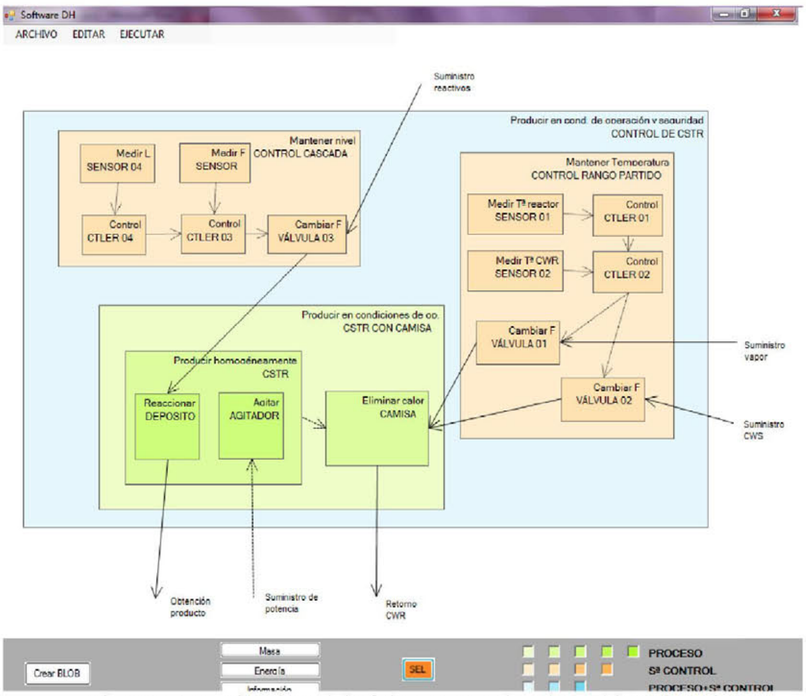

Figure 4. Functional model of the CSTR using the D-higraph tool

For the first test we have assumed that only a dynamic model is available for the reactor and the tank, both would receive actual input data from the plant. Using the reactor dynamic model a fault has been identified following De la Mata & Rodríguez (2011).

This fault has been input to the functional model, which generates a tree with the consequences of the fault. For the second test we assume that there is a complete model of the plant. In this case fault has been observed and the functional model has generated the tree of possible causes. This set of possible causes is simulated using the dynamic model in order to validate and quantify the deviations provided by the functional model.

5. Conclusions

In this paper an environment to use structural as well as functional models has been presented. Functional models are generated using the D-higraphs formalism and its exploitation is made using an expert system. Structural (steady state and dynamic) models are generated using the AspenTech software. Communication between both representations is made using a dll component which can control the simulation or generate events for the functional representation. This approach has been tested on a sample process with satisfactory results. It can be used in many different ways as to validate and quantify functional models, to validate sensors and alarms or to provide falt tolerant control (through control reconfiguration). Future research is being focused on automating all the procedure and on testing and creating a procedure for automatic control reconfiguration.

References

Álvarez, M. E. (2010). Diagnosis de fallos en procesos químicos mediante modelos D-higraph. *Final Project, Department of Chemical Engineering, Technical University of Madrid.*

CLIPS (2011). *CLIPS, A Tool for Building Expert Systems.* March, 13, 2011. http://clipsrules.sourceforge.net/

De la Mata, J. L., Rodríguez, M. (2010a). Abnormal Situation Diagnosis Using D-higraphs. *Proc. of the 20th European Symposium on Computer Aided Process Engineering (ESCAPE 20),* pp. 1477-1482.

De la Mata, J. L., Rodríguez, M. (2010b). D-higraphs. *ASLab Report.* http://www.aslab.org

De la Mata, J. L., Rodríguez, M. (2011). Self-learning of fault diagnosis identification. *Proc. of the 21th European Symposium on Computer Aided Process Engineering (ESCAPE 21),* pp. xxx-xxx.

Harel, D. (1987). Statecharts: A Visual Formalism for Complex Systems. *Sci. Comput. Program, 8 (3),* pp. 231-274.

Lind, M. (1994). Modeling Goals and Functions of Complex Industrial Plant. *Applied Artificial Intelligence, 8 (2),* pp. 259-283.

Morbach, J., Yang,A., Marquardt, W. (2007). OntoCAPE-A large-scale ontology for chemical process engineering. *Engineering Applications of Artificial Intelligence*, 20 (2), pp. 147-161.

Rodríguez, M., Sanz, R. (2009). Development of Integrated Functional-Structural Models. *Computer Aided Process Engineering, 27,* pp. 573-578.

Wiesner, A., Morbach, J., Marquardt, W. (2011). Information integration in chemical process engineering based on semantic technologies. *Computers & Chemical Engineering*, 55 (4), pp. 692-708