

Chapter 3

Description of Connectivity and Causality

Abstract In this chapter, we discuss the description of two related yet different notions—connectivity and causality. Connectivity shows a physical or information linkage between process units; this linkage illustrates qualitative process knowledge without using first-principle models. The main resources for establishing connectivity are process flow diagrams (PFDs) and piping and instrumentation diagrams (P&IDs); thus we need to convert them into standard formats, such as adjacency matrices, digraphs, and semantic web models, which are easily accessible and computer-friendly. Causality between process variables can be built through process data as well as process knowledge; thus it can be described qualitatively, yet sometimes with certain quantitative information, by structural equation models, matrices and digraphs, and matrix layout plots.

Keywords Process flow diagrams · Piping and instrumentation diagrams · Adjacency matrices · Reachability matrices · Directed graphs · Semantic web · Extensible markup language · Resource description framework · Web ontology language · Structural equation models · Matrix layout plots

We begin our discussion with the description of two related yet different notions—connectivity and causality. For each of them, there are multiple formats; we will show some typical ones.

3.1 Description of Connectivity

Connectivity shows a physical or information linkage between process units; this linkage illustrates qualitative process knowledge without the needs of first-principle models. The main resource for establishing connectivity are process flow diagrams (PFDs) and piping and instrumentation diagrams (P&IDs); thus we need to convert them into standard formats that are easily accessible and computer-friendly. In what follows we introduce three main formats for this purpose.

3.1.1 Adjacency Matrices

An adjacency matrix [6, 7] is a matrix form to express topology with directionality. This notion of adjacency stresses that only one-step or direct connectivity is included whilst the indirect relationship is excluded because it can be inferred.

For a system with n elements ($x_k, i = 1, \dots, n$), an $n \times n$ adjacency matrix \mathbf{A} can be defined. Each entry a_{ij} is binary: if element x_i is adjacent or directly connected to element x_j , then $a_{ij} = 1$; otherwise $a_{ij} = 0$.

Based on the adjacency matrix \mathbf{A} , another binary matrix, reachability matrix \mathbf{R} , can be derived to describe both direct and indirect relationships. Even if x_i is not adjacent to x_j , x_j may still be reached by x_i via other elements. If x_j is reached by x_i via a third element x_k , then it is called a 2-step reachability, to distinguish from the adjacency as the 1-step reachability. Similarly, k -step reachability can be defined. It can be proved that the k -step reachability can be described as the Boolean equivalent of \mathbf{A}^k , where the Boolean operator is defined as follows for each entry of the matrix:

$$(a_{i,j})^\sharp = \begin{cases} 1, & a_{i,j} \neq 0, \\ 0, & a_{i,j} = 0. \end{cases} \quad (3.1)$$

Thus, a reachability matrix is defined as:

$$\mathbf{R} = (\mathbf{A} + \mathbf{A}^2 + \dots + \mathbf{A}^n)^\sharp. \quad (3.2)$$

The summation is from 1 to n because it can be proved that if two elements are not reached from one to the other via n steps, then they cannot be reached via more steps. In matrix \mathbf{R} , each entry r_{ij} means whether x_i can reach x_j .

Take a tank system with cascade control as an example, as shown in Fig. 3.1. To show the adjacency between each pair of elements, such as the tank, pipes, and controller, an adjacency matrix can be constructed, as shown in Fig. 3.2a. By matrix computation, one can obtain the 2-step reachability $(\mathbf{A}^2)^\sharp$ as shown in Fig. 3.2b, 1- or 2-step reachability $(\mathbf{A} + \mathbf{A}^2)^\sharp$ as shown in Fig. 3.2c, and finally the reachability matrix \mathbf{R} as shown in Fig. 3.2d.

3.1.2 Digraphs

As an alternative of the adjacency matrix \mathbf{A} , when each element in the system is expressed by a node and each '1' entry is expressed by an arc linking two nodes corresponding to the two indices in \mathbf{A} , matrix \mathbf{A} is converted into a directed graph or digraph including n nodes. By this conversion, the connectivity is visualized and can be better understood due to its intuitivity, because this digraph simply shows the PFD or P&ID by converting each element into an abstract node. The connectivity of the above example can be described by the digraph as shown in Fig. 3.3. Based on

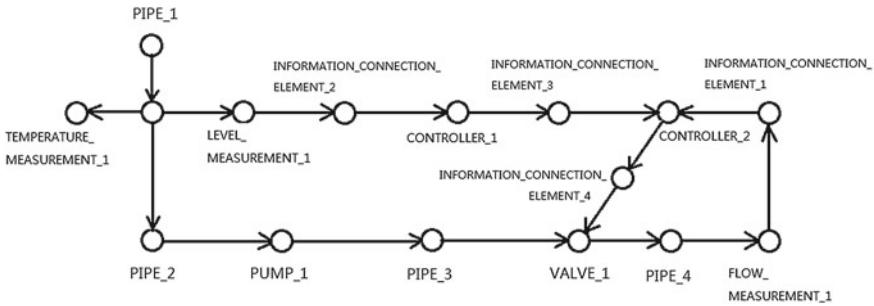


Fig. 3.3 Digraph of the tank system

The graph representation is particularly beneficial when the matrix is sparse. Moreover, some quantitative or dynamic factors can also be attached to the graph to extend its model description, which is useful in some application areas.

3.1.3 Semantic Web Description

In addition to the above mathematical descriptions, with the development of the Semantic Web, a new data model has come into use, namely, the ontology framework, which is based on a combination of artificial intelligence and database techniques. The ontology framework can be regarded as a conceptual model defined by a computer understandable language to describe and categorize the units/resources or linkages between units and their relationships. It translates the concepts defined and understood by humans into semantics in the cyber world defined by classes and rules. After this translation, new knowledge can be generated or discovered by machines through an automated inference, which makes the representation more powerful and useful [1]. By applying this technique to process modeling, the process connectivity can be modeled on the basis of PFDs or P&IDs, which facilitates the modeling and inferencing without using other special tools.

In terms of computer aided engineering exchange (CAEX) schema, eXtensible Markup Language (XML) gives users sufficient freedom to further define syntaxes and classes in their respective areas. An adjacency matrix can be constructed using the parsed information from its CAEX description—XML files [3, 9]. For the purpose of process topology description, however, a more uniform way is needed to define the process units (considered as resources) and their connections. The combination of Resource Description Framework (RDF) (<http://www.w3.org/RDF/>) and Web Ontology Language (OWL) (<http://www.w3.org/2004/OWL/>) provides a general method for conceptual description or modeling of information that is implemented in web resources, using syntaxes. In addition to connectivity, this ontological model can describe additional information such as constraints and conditions that are important for process modeling in an interoperable way.

Table 3.1 Classes of resources in the ontology framework

EQUIPMENT	UNCONTROLLED_ELEMENT	PIPE, TANK
	CONTROLLING_ELEMENT	AUTOMATIC_ELEMENT (PUMP, VALVE), MANUAL_ELEMENT (MANUAL_VALVE)
	MEASURING_ELEMENT	FLOW_MEASUREMENT, LEVEL_MEASUREMENT, TEMPERAUTRE_MEASUREMENT
COMPUTER	ANALOG_ELEMENT	CONTROLLER, SELECTOR
	DIGITAL_ELEMENT	AND, OR, NOT
	INFORMATION_CONNECTION_ELEMENT	
	AUTOMATIC_ELEMENT	PUMP, VALVE
	MEASURING_ELEMENT	FLOW_MEASUREMENT, LEVEL_MEASUREMENT, TEMPERAUTRE_MEASUREMENT

Based on the needs of process control, we first define resources by classes, which can be divided into two groups: one is equipments in the physical world, including process units and instruments; the other is computers or processors in the cyber world. Some resources can belong to both worlds, resulting in the coexistence in the two groups. From the control system perspective, sensors (transmitters), controllers, and actuators should be included in the latter category; while the sensors and actuators should also be contained in the former category because they are physical equipment. The relationship between these resources in the class domain is inheritance, namely, a subclass under a class inherits all the properties of the class; of course, a class can belong to multiple classes and inherit all the properties from them. For the tank system, a list of classes is shown in Table 3.1. Note that both the physical linkage, PIPE, and the information linkage (signal line), INFORMATION_CONNECTION_ELEMENT, are defined as classes. Next, properties are assigned to resources; these resources are the subjects of the properties. In addition to datatype and annotation properties, we define the following object properties to describe the physical and information linkages:

- `uncontrolledElement.measuringElement`: linkage from an uncontrolled element to a measuring element, e.g., the level of a tank measured by a sensor.
- `uncontrolledElementOutlet.uncontrolledElementInlet`: linkage from an uncontrolled element to another uncontrolled element, e.g., a tank connected to a pipe as an outlet.
- `uncontrolledElementOutlet.controllingElementInlet`: linkage from an uncontrolled element to a controlling element, e.g., a pipe connected to a control valve.
- `controllingElementOutlet.uncontrolledElementInlet`: linkage from a controlling element to an uncontrolled element, e.g., a valve connected to a pipe.
- `computer.computer`: linkage from a computer to another computer, e.g., a controller connected to a signal line (information connecting element).

The domain and range of the properties should be defined as appropriate resources.

For the tank system example (Fig. 3.1), to build the OWL file, we add instances of the above defined classes. Properties are assigned to them to define the contents and inter-relationships. For example, the outlet of PIPE_1 is connected to TANK_1; hence PIPE_1 has an object property, which is uncontrolledElementOutlet.uncontrolledElement, to have the value of another instance, TANK_1. The ontology can be visualized by OntoViz[®], a plug-in for Protégé-OWL[®], as shown in Fig. 3.4.

To query ontology-based RDF/OWL files, SPARQL Protocol and RDF Query Language (<http://www.w3.org/TR/rdf-sparql-query/>) can be used to capture useful information and conduct inferences. SPARQL uses query triples as expressions with logic operations such as conjunctions and disjunctions to perform inferences based on semantics.

One can use SPARQL to test connectivity based on object properties. If one defines a general object property and regards all the other object properties including physical and information linkages as its subproperties, then the connectivity with specified steps can be obtained. Moreover, by defining the object property as transitive, a measure of reachability can be obtained directly to show the domain of influence triggered by a change in one object.

3.2 Description of Causality

In addition to connectivity, causality between process variables should also be described. Note that the modeling resources herein include process data as well as process knowledge.

3.2.1 Structural Equation Models

Structural equation modeling (SEM) is a statistical technique for testing and estimating causal relations [8, 10]. A structural model shows potential causal dependencies between endogenous/output and exogenous/input variables, and the measurement model shows relations between latent variables and their indicators. For example, if an endogenous variable y is influenced by exogenous variables x_1 and x_2 (assume that all variables are normalized to have zero mean and unit variance), a regression model can be built as $y = p_{y1}x_1 + p_{y2}x_2 + p_{y\epsilon}\epsilon$ and thus be depicted as a path diagram in Fig. 3.5, where each parameter p is called a path coefficient, and ϵ represents the residual, that is, collective effect of all unmeasured variables that could influence y . The directed arrows represent the influence of the exogenous variables and the residual on the output variable, and the bidirectional arrow represents the correlation between exogenous variables.

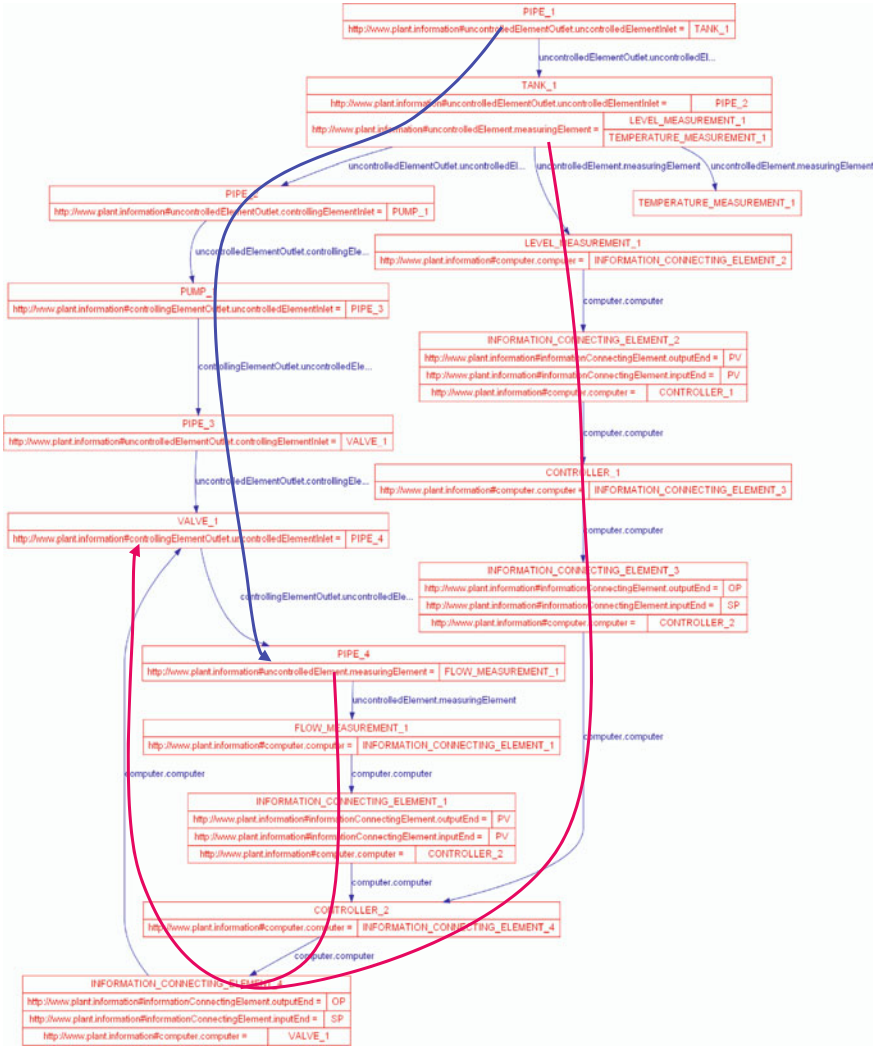
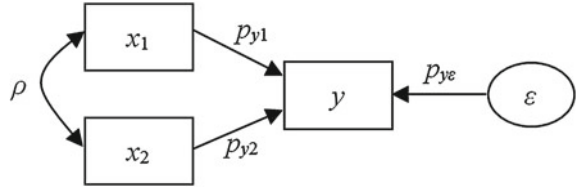


Fig. 3.4 Visualization of the tank system ontology exported by Ontoviz[®] with material flow (blue) and information flow (red) paths

This model is a statistical model and is highly dependant upon the partition of variables. What is more important is to obtain the topology of the system, where each variable can be both input and output variables. Thus we usually use the following descriptions.

Fig. 3.5 Path diagram of a structural model



3.2.2 Matrices and Digraphs

Similar to the matrix or digraph formats to describe connectivity, these models can also be used to describe causality. We have mentioned that we can introduce other information onto the arcs of a digraph model. Typically signs can be attached to the arcs to describe positive (promotion) or negative (inhibition) relation. For example, if the increase (decrease) of variable x_i can cause the increase (decrease) of variable x_j , then we define the sign as '+'. On the contrary, if the increase (decrease) of variable x_i can cause the decrease (increase) of variable x_j , then we define the sign as '-'. This model is called a signed digraph or signed directed graph (SDG). Normally we use solid and broken lines to denote positive and negative relations respectively. The formal definitions are as follows [4, 5, 11, 12]:

Definition 1: A SDG γ is the composite (G, φ) of

- (i) digraph G_0 that is the quadruple $(N, A, \delta^+, \delta^-)$ of
 - (a) a set of nodes $N = \{n_1, n_2, \dots, n_m\}$,
 - (b) a set of arcs $A = \{a_1, a_2, \dots, a_n\}$,
 - (c) a couple of incident relations $\delta^+ : A \rightarrow N$ and $\delta^- : A \rightarrow N$ that map each arc correspond to its original node and terminal node, respectively, and
- (ii) a function $\varphi : A \rightarrow \{+, -\}$, where $\varphi(a_k)$ ($a_k \in A$) is called the sign of arc a_k .

Definition 2: A pattern on the SDG model $\gamma = (G, \varphi)$ is a function $\Psi : N \rightarrow \{+, 0, -\}$. The value $\Psi(v)$ ($v \in N$) is called the sign of node v , i.e.

$$\begin{aligned} \Psi(v) &= 0, & \text{for } |x_v - \bar{x}_v| < \varepsilon_v, \\ \Psi(v) &= +, & \text{for } x_v - \bar{x}_v \geq \varepsilon_v, \\ \Psi(v) &= -, & \text{for } \bar{x}_v - x_v \geq \varepsilon_v, \end{aligned}$$

where x_v is the measurement of the variable v , \bar{x}_v is the normal value, and ε_v is the threshold.

Definition 3: Given a pattern Ψ on a SDG model $\gamma = (G, \varphi)$, an arc a is said to be consistent(with Ψ) if $\Psi(\delta^+a) \varphi(a) \Psi(\delta^-a) = +$. A path, which is consisted of arcs a_1, a_2, \dots, a_k linked successively, is said to be consistent (with Ψ) if $\Psi(\delta^+a) \varphi(a_1) \dots \varphi(a_k) \Psi(\delta^-a) = +$.



Fig. 3.6 SDG of the tank system with level control

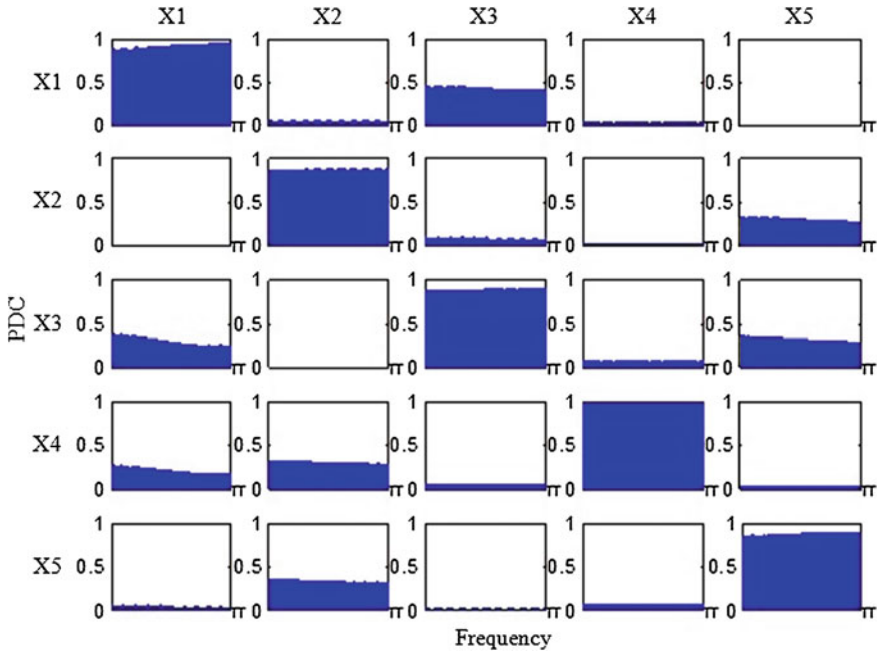


Fig. 3.7 Example of a matrix layout plot

Recall the tank system example. This time we only focus on the level control and related variables—inlet flow rate (F_1), outlet flow rate (F_2), and liquid level (L). When the level is high, the valve will open to increase the outlet flow rate according to the control law, and the result is the reduction of the level. Thus the SDG is as shown in Fig. 3.6.

The graph model is the main description of causality and we will discuss the modeling approaches and applications in the following chapters.

3.2.3 Matrix Layout Plots

Although causality is a qualitative description, it is often captured through quantitative data analysis, leading to additional information. A typical method is partial directed coherence (PDC), which has been developed and used in the neuroscience area [2]. This method can be used for multivariate systems to extract the direct causality between each pair of variables.

In the frequency domain analysis, matrix layout plots are often used, as shown in Fig. 3.7 (for details see Chap. 4). Each plot shows the information transfer from one variable to another. It is to be noted that the cause variables are listed on the top while the effect variables to be tested are on the left, which is not the same with the matrix forms mentioned earlier.

3.3 Chapter Summary

Model description is the basis of all kinds of analysis. Thus various descriptions of connectivity and causality have been briefly introduced in this chapter, which will be discussed in detail in the next chapters via different modeling approaches and other applications.

The descriptions in this chapter are limited to mathematical models and ontology models; they can be understood by computers as well as humans. The benefit is that they have potential to automate the modeling and analysis procedures. The ontology work is still ongoing, but this description has many advantages and conforms to World Wide Web Consortium (W3C) recommendations.

References

1. Allemang D, Hendler J (2011) *Semantic web for the working ontologist: effective modeling in RDFS and OWL*, 2nd edn. Morgan Kaufmann Publishers, San Francisco
2. Baccala LA, Sameshima K (2001) Partial directed coherence: a new concept in neural structure determination. *Biol Cybern* 84(6):463–474
3. Fedai M, Drath R (2005) CAEX—a neutral data exchange format for engineering data. *ATP Int Autom Technol* 3(1):43–51
4. Iri M, Aoki K, O'shima E, Matsuyama H (1979) An algorithm for diagnosis of system failures in the chemical process. *Comput Chem Eng* 3(1–4):489–493
5. Iri M, Aoki K, O'shima E, Matsuyama H (1980) A graphical approach to the problem of locating the origin of the system failure. *J Oper Res Soc Jpn* 23(4):295–311
6. Jiang H, Patwardhan R, Shah SL (2009) Root cause diagnosis of plant-wide oscillations using the concept of adjacency matrix. *J Process Control* 19(8):1347–1354
7. Mah RSH (1989) *Chemical process structures and information flows*. Butterworth, Boston
8. Pearl J (2009) *Causality: models, reasoning, and inference*, 2nd edn. Cambridge University Press, Cambridge
9. Thambirajah J, Benabbas L, Bauer M, Thornhill NF (2009) Cause-and-effect analysis in chemical processes utilizing XML, plant connectivity and quantitative process history. *Comput Chem Eng* 33(2):503–512
10. Wright S (1921) Correlation and causation. *J Agric Res* 20:557–585
11. Yang F, Xiao D (2005) Review of SDG modeling and its application. *Control Theor Appl* 22(5):767–774
12. Yang F, Xiao D, Shah SL (2010) Qualitative fault detection and hazard analysis based on signed directed graphs for large-scale complex systems. In: Zhang W (ed) *Fault Detection*. In-Tech, Vukovar, Croatia, pp 15–50