8

Conclusions

The research in this monograph has presented a distributed approach to the control of process operations that require a high degree of reconfigurability. A distributed coordination approach based on the distributed paradigms of holonic manufacturing and supply chain management was developed in this text to develop a blueprint for reconfigurable process control systems.

In this chapter we now summarise the key contributions from this work, the limitations of the research, and the areas for future work where this research can be extended.

8.1 Main Contributions

The DRPC approach promotes a bottom-up method to the design and integration of process control systems. The bottom-up approach is preferred as it enables rapid integration and reconfiguration, both during and after the design life-cycle. The overall design method of a DRPC system then operates in the following sequence:

i. a top-down decomposition of the top-level requirements into corresponding low-levels requirements;
ii. interpretation of the low-level requirements into the selection of process elements and assignment of their control responsibilities; and
iii. design, implementation, and integration of process elements into a complete system in a bottom-up manner.

While the top-level requirements in step (i) are expected to cover a range of possible production scenarios, they may not – and need not be – exhaustive as the design should be reconfigurable enough to allow for new requirements at any stage in the design or operation life-cycle. Similarly, the decomposition of top-level requirements may only be required to the level of abstraction where they can be delivered by the self-contained design of process elements.

For example, in case of a reaction operation, if the design of the reactor element permits, the low-level requirement assigned to that element should be of the form ⟨perform reaction 'X'⟩ as opposed to specifying in detail the operation of individual actuators or the control policies. In steps (ii) and (iii), it remains important that the principle of *low and least commitment* (Valckenaers & van Brussel 2005) is employed so as to induce a maximum level of flexibility between elements to operate over a range of conditions, both planned or unplanned.

Within this framework, the developments in Chapters 4, 5 and 6 provide the concepts and necessary guidelines to develop a design process from which an RPC system can be developed. These developments can be summarised as below.

Distributed Control Architecture

The description in Chapter 4 started with introducing a new concept of *process element* as a stand-alone, modular building block of a DRPC system. It was suggested that the identification of process elements is done based on their physical involvement in the process while also observing that each element must have at least one but possibly more decisions that it can regulate on its own. A systematic method to perform such an identification was developed by Bussmann (Bussmann, Jennings & Wooldridge 2001, Jennings & Bussmann 2003, Bussmann 2003) in the context of a discrete process.

Chapter 4 later also identified and defined the structure of four key process element types as forming any DRPC system. When engineering a DRPC system, this identification of element types should be used to develop a library of multipurpose designs of process elements that can be deployed as 'off-the-shelf'. In step (ii) in the overall design method discussed at the beginning of the section, an appropriate element can be selected from this library to meet the low-level requirements.

Distributed Interaction Model

Chapter 5 introduced a distributed model for managing the interactions between process elements. The proposed model builds upon two key aspects: (a) the supplier-customer design of process elements is used based on which these elements acquire their feedstocks and services from respective supplier elements and (b) the demand-pull type interaction behaviour is used to build the plantwide process schemes. The five steps of the reconfiguration process (Fig. 5.1) then define the sequence of interactions that elements must follow to develop or reconfigure a process scheme in response to changing plant conditions. The interaction model also characterised product-centric and unit-centric approaches for recipe mapping as the two distinct approaches, the former being more appropriate for high variety of products while the latter more appropriate for frequent changes between the same products.

Distributed Coordination Algorithm

One particular aspect of the interaction model – that of identifying the local operating settings of unit elements – was considered in Chapter 6 as an example of developing a distributed coordination strategy for managing the localised operations of process elements. It was shown that an economic interpretation of so-called *nested decomposition* algorithm can provide a systematic method to implement the intended demand-pull type, price-demand guided interactions between elements in a mathematical form. While the classical techniques in nested decomposition (Ho & Manne 1974, O'Neill 1976, Wittrock 1985) apply generally to series-connected networks, their extension using so-called *approximate cut update* technique enables their use in process networks of arbitrary but acyclic nature. The implementation of the algorithm is centered around a single, general-purpose unit module (see Appendix B) that covers all possible combinations of connections in which a unit element can be located within an acyclic network. For a practitioner intending to deploy the algorithm (even as part of conventional hierarchical model), it thus suffices to develop a single such module that can be incorporated as part of the design of any unit element.

8.2 Limitations of the Research

This work forms one of the first attempts at using distributed coordination for developing reconfigurable process operations. Since the scope of the development is wide, naturally some limitations remain. Below we describe key such limitations where further work could be useful.

- *Organisation of process elements:* The control architecture in Chapter 4 was developed to the extent of identifying the types of process elements and defining their structure, *i.e.,* data models, control functions and connections. An account of the flexibility in terms of the alternative configurations in which they can be organised (when developing a reconfigurable process plant from grass route) or the existing configurations can be changed (when revamping an existing control system) was not discussed at length as this forms a question of practical implementation. Some examples of alternative choices were given in Chapter 7, however for more specific design practices, it will be necessary to quantify how much flexibility is sufficient and cost-effective so as to address the end-user needs for a foreseeable future.
- *Interaction behaviour of header and service elements:* The discussions in Chapters 5 and 6 can be extended to cover the interactions of header and service elements. For header elements, the issue is to define systematic methods for reconfiguring the process routes in a coordinated manner especially when the transients occur. For service elements, the methods for

service distribution must be linked closely with the material exchange interactions because the consumer elements which use these services may also be connected via physical routes and therefore change in service allocation at one point can affect the operations in other parts of the network.

- *Nonlinear dynamics, transients and recyles in distributed coordination:* The distributed control problem used as a basis in Chapter 6 was limited to a simplified problem based on linear, steady-state dynamics model. The solution strategy developed therein could be generalised to other class of problems involving non-linear and/or dynamical models – the former class of problems can arise in distributing the optimisation layer while the latter in distributing the advanced control layer.

8.3 Future Challenges

In addition to the above limitations, there remain other broader issues concerned with this research where further challenges remain.

- *Tools for human interactions:* Tools for human interactions should be developed to define where and how a human role is involved as part of the reconfiguration process. Inclusion of the human role will be important in DRPC in identifying that a reconfiguration is necessary, and defining a feasible configuration from the available choices. It is envisaged that the actual reorganisation of elements will be automated in future, but humans will play a key role in making this happen.
- *Design methodology and integration within industrial practice:* A comprehensive design method for DRPC should be developed that offer complete guidelines for developing a system that fulfills end-user requirements for a sufficiently foreseeable future. The method should be also detailed enough to enable an engineer not familiar with distributed concepts to perform design tasks with little or no external help. For the short-term future, the design method should also consider a migration strategy for operators of the existing plants to incrementally move towards building the DRPC system envisioned here. Some aspects of migration have been discussed in Section 4.3.
- *Management of virtual enterprises:* Finally, the most closely related field to the research in this text is the field of managing virtual enterprises themselves. In the past, supply chain management has benefited from control engineering tools, *e.g.*, for studying inventory control (Deonckheere, Disney, Lambrecht & Towill 2002, Perea-López, Grossmann, Ydstie & Tahmassebi 2000, Chaib-draa & Müller 2006). The distributed coordination strategy developed in Chapter 6 could be extended to solve the large-scale control problems associated with managing a virtual enterprise in a distributed manner.