Hierarchical Colored Petri Nets Based Components for Workflow Systems

Feng Zhou, Rui-Feng Bo
School of Mechanical Engineering, Dalian University of Technology,
Dalian, Liaoning, 116023, P.R. China
Hong-Zhong Huang*, Dan Ling, Qiang Miao
School of Mechatronics Engineering, University of Electronic Science and Technology of China,
Chengdu, Sichuan, 610054, P.R. China

Workflow systems have benefited the enterprise and customer in many aspects. But with the increasing complexity of the system, workflow design becomes a complicated and time-consuming process. In this paper, we model the system based on the Hierarchical Colored Petri Nets (HCPN) to avoid the complexity of the system, and the super nets of the model are abstracted as independent components to increase the flexibility, acceptability and maintainability of the system. Another important characteristic of this model is its convenience for reuse and workflow mining. The development process of the overhead traveling crane was given to demonstrate the proposed method.

Key Words : Workflow, Component, Hierarchical Colored Petri Nets, Workflow Mining

1. Introduction

Today's business environment makes it difficult for a company to deal with business processes without the support of the workflow. A workflow is the computerized facilitation or automation of a business process, in whole or part. The core of workflow system is the notion of business process, which involves a set of ordered activities to be undertaken by human or other resources in the enterprise in order to meet the predefined requirements. Many enterprises have benefited from workflow-based applications, like flexibility, integration and reusability (Salimifard et al., 2001 ; Worley et al., 2002).

The specialists and scholars in the world have done many researches on workflow systems. Most of the researches are summarized as follows.(1) some application-relevant factors were incorporated into the model to extend the workflow models such as time restriction, resources scheduling ; (2) the workflow model standard or a common language were updated or established to improve the capability of interoperation and interchange between the workflow systems ; (3) semantic researches such as checking the deadlock and the loop of workflow execution, workflow equivalence and workflow optimization were done to ensure the execution of the workflow ; (4) the workflow application system development and application-relevant researches, e.g., agent-based workflow, workflow interoperation, and enhancing the quality and productivity of large workflow systems development (Zhuge, 2003).

With the increase of system complexity, workflow design becomes a complicated and time-consuming process. To reduce the complexity of the system, we introduced the theory of HCPN to model system. The main descriptions and models of the processes in the workflow system are de-

* Corresponding Author,
E-mail : hzhuang@uestc.edu.cn
TEL : +86-28-83206916; **FAX** : +86-28-83206916
School of Mechatronics Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan 610054, China. (Manuscript **Received** May 4, 2005; **Revised** July 4, 2006)

scribed by basic net. The details and additional information are described by super nets. To reduce lead time and improve quality, reuse from previous successful workflow models is the most promising method. So we model the system based on the HCPN and abstract some super nets as independent components for the reuse of the system and workflow mining.

The research scope of this paper focuses on the methodology of mining and reusing the workflow systems. The rest of the paper is organized as follows. The concepts of the workflow, component and Hierarchical Colored Petri Nets are introduced in Section 2 and Section 3. The architecture of our model is then described in Section 4. The development process of the overhead traveling crane as an example is given in Section 5. Finally, conclusion and future work are presented in Section 6.

2. Workflow Components

2.1 Workflow

The concept of workflow has been developed from the notion of factory automation in 1970s, when the first prototype was developed. During the 1970s and 1980s, some workflow applications were not successful due to unavailability of sufficient technology at that time. Since the late 1980s, with the development of technology, workflow has become a motivating and challenging research field. And many research achievements were gained during that time (Salimifard et al., 2001).

The workflow management coalition defines a workflow as a computerized facilitation or automation of a business process, in whole of part. A specific software system, so-called WfMS, controls its automated aspects. Based on the level of persistence in process definition and routing of tasks, workflows are classified as production workflow, administrative workflow, and special workflow. No matter which type of these workflows, the process of modeling is far from trivial: It requires deep knowledge of the process and workflow language. So constructing workflow models from scratch is a difficult and error prone task. It often requires the participation of the specialists and experts (Schimm, 2004).

The workflow life cycle consists of four phases as shown in Figure1 : (A) workflow design, (B) workflow configuration, (C) workflow execution, (D) workflow diagnosis. In the design phase, the workflow is constructed by specialists or experts. If design is finished, the user has to deal with the limitation and particularities of the workflow models during the configuration phase. Workflow instances are handled by the workflow system as specified in the design phase and realized in the configuration phase during the execution phase. The diagnosis phase can provide input for another design phase thus completing the workflow life cycle. In the traditional approaches, less attention is paid to the execution and diagnosis phase, so much valuable and important runtime data as input for redesign would be typically missed (Van der Aalst et al., 2003).

Reuse is a key factor for decreasing the time needed to develop an application and improving its quality, and this concept can be applied at different levels (specification, code lines, components, software, etc.) (Worley et al., 2002). The extraction of workflow models from those runtime data and logs is called workflow mining. The goal of workflow mining is to reverse the process and collect them to support workflow design and analysis. Some workflow mining theories, algorithms and tools have been established by scholars and

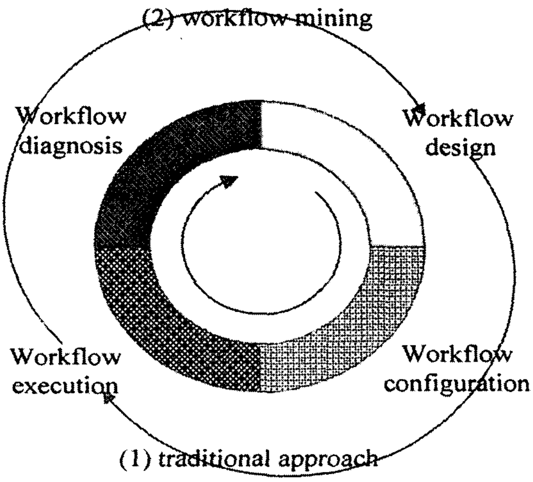

Fig. 1 Life cycle of workflow models

experts in the corresponding fields (Van der Aalst et al., 2004; Herbst et al., 2004; Pinter et al., 2004). But most of those mining models are helpless before the complex process models and vast varied data. In this paper, we established some workflow components for the reuse of workflow system.

2.2 Workflow components

A component is a reusable software package or application. A standard application has two main parts, the implementation and the data. The interaction between two applications is through the database. In fact this interaction is limited to the data sharing. A component-based application provides operation services containing the data operation and method operations in its implementation (Rosenman et al., 2001).

As important parts in the workflow system, workflow components should have these characteristics described as follows (Zhuge, 2003). Firstly, a workflow component should be independent, which means one component can be executed independently and doesn't have common tasks with another component. Secondly, a component should be encapsulated so that it can be used, executed, and viewed as a whole like a single task. This needs an access interface to normally specify the input and output of the component in the workflow systems. Thirdly, a workflow component should satisfy internal process completeness (definition completeness and execution completeness). Finally, a component should satisfy the internal consistency, such as data type consistency and temporal consistency.

A workflow system is composed of two main modules: the build-time module and the run-time module (Salimifard et al., 2001; Bae et al., 2002; 2004, Zhang et al., 2006). The build-time module is the workflow process definition module, which is used in build-time to generate a computerized definition of the system process. It provides system modeling tools and helps the designer to design, test, simulate, optimize and validate the workflow process. Precise specification of real system's module has a tendency to become large and complex. To meet the modeling requirement, we establish the build-time module based on the Hierarchical Colored Petri Nets and some typical structures are abstracted into workflow components for the mining and reuse. The run-time module is the workflow execution module, which controls, coordinates and records the workflow execution during run-time. Some functional components were built in the run-time module to perform their duties respectively, such as process manager, document manager and task manager.

All components are stored and managed in the workflow components repository for the reuse purpose. A workflow component is specified in the repository by the following frame:

```
ComponentName :: ⟨Name⟩ ;
  ReuseType : ⟨Type⟩ ;
  Process : ⟨NoteSet, FlowSet⟩ ;
  InType :. ⟨Type⟩ ;
  OutType : ⟨Type⟩ ;
  Restriction : ⟨Representations⟩;
  InDataStru : ⟨Structure⟩ ;
  OutDataStru : ⟨Structure⟩ ;
  ExeDataStru : ⟨Structure⟩ ;
  ......
```

3. Definition of HCPN

Petri nets are widely studied and successfully applied in different discrete-event dynamic systems (Jensen, 1994; Huang et al., 2004; Huang et al., 2005). Their strong mathematical foundation and the availability of a wide range of supporting tools made them popular among the researchers. Colored Petri nets have got their name because they allow the use of tokens that carry data values and hence be distinguished from the low-level Petri nets. Although using classical colored Petri nets will make the description and analysis of the system model more compact and amenable, we need new extension of the modeling technology. The hierarchy structure was introduced into the colored Petri nets, called HCPN. We could only describe the main descriptions and models of the process in one level without considering the details and additional information which will be described in detail in super nets. This is the exact reason for us to introduce the hierarchical theory to the workflow model.

A HCPN is defined as a tuple, which gives a mathematically sound and unambiguous definition of CPN and its semantics. It has a set of substitution transitions which allow the embedding of small subnet into the basic net for another level modeling. We call the non-hierarchical CPN basic net and the input or output places of the net port nodes. The port nodes specify and check the type and number of tokens that can be introduced into the net. The substitution transition is called a super node and the nets that contain them are called super net. In a super net, the places that are connected to the super nodes (substitution transitions) are called socket nodes. The socket nodes of a super node (substitution transition) are related to the port nodes of the net through the port assignment function.

The set of all elements in T is denoted by the type name T itself. The type of a variable, v, denoted by Type(v). The type of an expression, expr, denoted by Type(expr). The set of variables in an expression, expr, denoted by Var(expr). A binding of a set of variables, V, associating with each variable $v \in V$ an element $b(v) \in \mathrm{Type}(v)$. The value obtained by evaluating an expression, expr, in a binding, b, denoted by $\mathrm{expr}\langle b\rangle$. Var (expr) is required to be a subset of the variables of b, and the evaluation is performed by substituting for each variable $v \in \mathrm{Var}(\mathrm{expr})$ the value $b(v) \in \mathrm{Type}(v)$ determined by the binding.

We use B to denote the Boolean type: {false, true}. Moreover, we extend the notation Type(v) to $\mathrm{Type}(A) = \{\mathrm{Type}(v) \mid v \in A\}$ where A is a set of variables.

Definition 1: A HCPN is a tuple CPN=(PG, Σ, P, T, A, N, C, G, E, I) where:

PG is a finite set of nets such that: (a) each net $pg \in PG$ is a non-hierarchical CPN and, (b) none of the nets has common net element with another;

Σ is a finite set of non-empty types also called color sets;

P is a finite set of places. It consists of ordinary places, port places and socket places;

T is a finite set of transitions;

A is a finite set of arcs such that: $P \cap T = P \cap A = T \cap A = \varnothing$;

N is a node function. It is defined from A into $P \times T \cup T \times P$;

C is a color function. It is defined from P into Σ;

G is a guard function. It is defined from T into expressions such that:

$$\forall t = T: [\mathrm{Type}(G(t)) = B \wedge \mathrm{Type}(\mathrm{Var}(G(t))) \subseteq \Sigma],$$
$$B = \{\mathit{true}, \mathit{false}\}$$

E is an arc expression function. It is defined from A into expressions such that:

$$\forall a \in A: [\mathrm{Type}(E(a)) = C(p)\mathrm{MS} \wedge \mathrm{Type}(\mathrm{Var}(E(a))) \subseteq \Sigma];$$

I is an initialization function. It is defined from P into closed expressions such that:

$$\forall p \in P: [\mathrm{Type}(I(p)) = C(p)\mathrm{MS}]$$

In our model, transition marked with an S-tag means this transition is a substitution transition and it would be replaced by a super net.

4. System Architecture

Figure 2 shows the workflow system architecture established based on HCPN for reuse of workflow components. It has two main parts: build-time modules and run-time modules. The figure also shows the procedures of establishing the system model.

As we all know, the experienced developers can use past experience to solve new problems. But different developers have different experiences

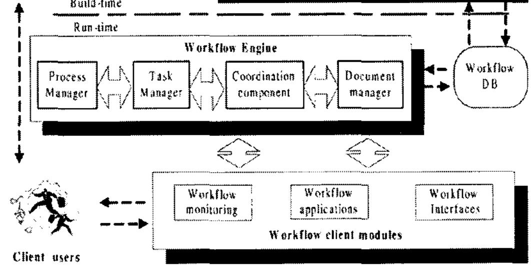

Fig. 2 Architecture of the workflow system

and skills as well as different understandings of the same work. The different understandings can lead to different workflow system and then different components are reused. So abstraction of the real system is a very important step to the developer who has understood and analyzed the real system. The abstraction can guide the designer to model proper system structure to reuse existing workflow components. Some special workflow components may be abstracted and designed to meet the new requirements of the new problem. These new components should be appended to the repository after being designed and used. To ensure the successful execution of the workflow system, we must composite whole system which means to composite the using components with the basic nets and relative components. There are several important factors we must adjust or connect, to accomplish the process and document design, they are: data structure, access interface and time or resource restriction. The work introduced above should be done during build-time.

Run-time modules are the workflow execution modules. The core of the run-time functions is the workflow engine, which controls, coordinates and records the workflow execution by several functional components. Process manager is responsible for controlling the execution order in terms of predefined process and managing all the process instances. Task manager is responsible for managing all the task instances. Coordination components coordinate the system by the coordination mechanisms established before to avoid deadlock and so on. Document manager is responsible for the creation, configuration and version management of the documents.

Workflow client modules provide the interfaces to the end user, include workflow monitoring, workflow application and so on. Client users could monitor the workflow process, forecast the results of the system through the monitoring interface; document processing interface provides the advantages that user can create or modify an actual document.

5. Application

In this section, the process of designing an overhead traveling crane is presented as an example to demonstrate the proposed system architecture. Figure 3 shows the structure decomposition of the generic overhead traveling crane. As we all

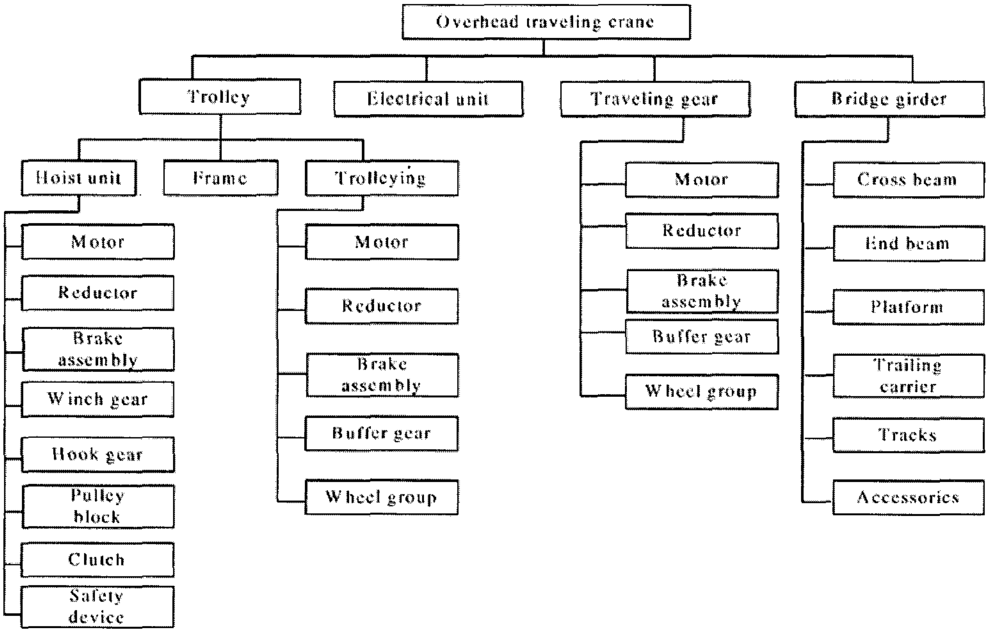

Fig. 3 Structure decomposition of traditional overhead traveling crane

know, design phase can be decomposed into three phases: conceptual design, technical design and working design. It is a very complicated work considering the dimensional and functional restrictions between those parts, coordination of participants and machine during the development. Dimension of the Cross beam and End beam will affect structure, dimension and selection of other parts and assemblies, such as frame of trolley, platform, trailing carrier, track, wheel groups, winch gear and etc.

To decrease the complexity of the model, we establish the workflow system based on the Hierarchical Colored Petri Nets. Figure 4 shows the workflow of the process of design bridge girder. The transition in the figure which is marked with the S-tag means it is a substitution transition and can be substituted by a corresponding super net to describe the detailed information of the transition represented. Figure 5 shows the super net which describes the technical design of the bridge girder. We give the simple description of the model on the basic net without considering the details which would be specified detailed on super nets. Some typical structures would be abstracted and made into components. These components would be used in basic net and be stored into the workflow component repository for reuse.

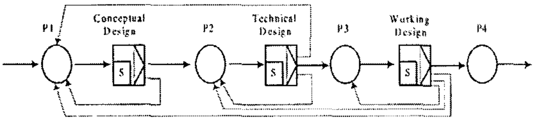

Fig. 4 Design phase of bridge girder

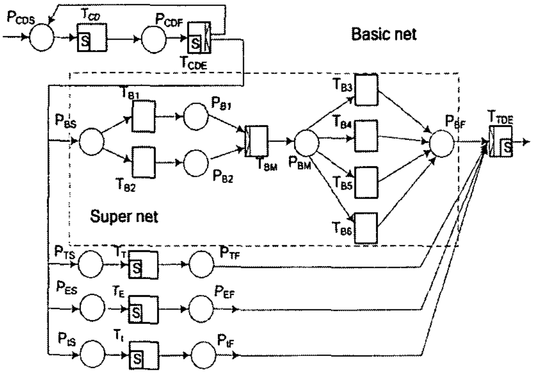

Fig. 5 Super net of the substitution transition (technical design of bridge girder)

Functional components are used from the repository during the system run-time. Process manager, task manager and coordination components ensure the workflow execution. Document manager plays an important act in the design system. Creation and modification of the document in the system are under the control of the document manager. These components also have the logs of the system execution. Engineers of the design group could monitor the workflow system by the client modules. The interfaces of the system help engineers manage the system process and get the documents of the crane.

Compared with traditional workflow systems, the proposed model has several advantages as follows:

Decrease the complexity of workflow system. Introduction of the HCPN and component make the complexity of the system hidden at the super net and components.

Convenient for the workflow mining and reuse. Theory of HCPN has a strong support to the workflow mining and components can be safely reused by any other workflow system.

Easy of run-time maintenance. The modeling mechanism could help the designers to localize the system errors.

Increase flexibility and acceptability. The change of the problem will lead to little change of the model compared with traditional workflow systems. The model can be more easily understood and used by different levels of users.

6. Conclusion and Future Works

The increasing complexity of the product development and design process increases the difficulty of modeling workflow system. The theory of HCPN and components are introduced to support the modeling of the system. The method increases the flexibility, acceptability and maintainability of the system and is convenient for the reuse and workflow mining. The development process of an overhead traveling crane was presented as the example to demonstrate the proposed method and system architecture.

In the future, we want to append the new rules into the coordination components and document manager to enhance their abilities.

References

Bae, H., Hu, W., Yoo, W. S., Kwak, B. K., Kim, Y. and Park, Y. T., 2004, "Document Configuration Control Processed Captured in a Workflow," *Computers in Industry*, Vol. 53, No. 2, pp. 117~131.

Bae, H. and Kim, Y., 2002, "A Document-Process Association Model for Workflow Management," *Computers in Industry*, Vol. 47, No. 2, pp. 139~154.

Herbst, J. and Karagiannis, D., 2004, "Workflow Mining with InWoLvE," *Computers in Industry*, Vol. 53, No. 3, pp. 245~264.

Huang, H. Z., Zhou, F. and Zu, X., 2005, "Petri Nets Based Coordination Component for CSCW Environments," *Journal of Mechanical Science and Technology*, Vol. 19, No. 5, pp. 1123~1130.

Huang, H. Z. and Zu, X., 2004, "Hierarchical Timed Colored Petri Nets Based Product Development Process Modeling," *Lecture Notes in Computer Science*, Vol. 3168, pp.378~387.

Jensen, K., 1994, "An Introduction to the Theoretical Aspects of Colored Petri Nets," *Lecture Notes in Computer Science*, Vol. 803, pp. 230~272.

Pinter, S. S. and Golani, M., 2004, "Discovering Workflow Models form Activity's Lifespans," *Computers in Industry*, Vol. 53, No. 3, pp. 283~296.

Rosenman, M. and Fu-jun W., 2001, "A Component Agent Based Open CAD System for Collaborative Design," *Automation in Construction*, Vol. 10, pp. 383~397.

Salimifard, K. and Wright, M., 2001, "Petri Net-Based Modelling of Workflow Systems: An Overview," *European Journal of Operational Research*, Vol. 134, No. 3, pp. 664~676.

Schimm, G., 2004, "Mining Exact Models of Concurrent Workflows," *Computers in Industry*, Vol. 53, No. 3, pp.265~281.

Van der Aalst, W. M. P., Van Dongen, B. F., Herbst, J., Maruster, L., Schimm, G., Weijters, A. J. M. M., 2003, "Workflow Mining: A Survey of Issues and Approaches," *Data & Knowledge Engineering*, Vol. 47, No. 2, pp. 237~267.

Van der Aalst, W. M. P., Weijters, A. J. M. M., 2004, "Process Mining: A Research Agenda," *Computers in Industry*, Vol. 53, No. 3, pp. 231~244.

Worley, J. H., Castillo, G. R., Geneste, L., Grabot, B., 2002, "Adding Decision Support to Workflow Systems by Reusable Standard Software Components," *Computers in Industry*, Vol. 49, No. 1, pp. 123~140.

Zhang, X., Huang, H. Z. and Yu, L., 2006, "Fuzzy Preference Based Interactive Fuzzy Physical Programming and Its Application in Multi-objective Optimization," *Journal of Mechanical Science and Technology*, Vol. 20, No. 6, pp. 731~737.

Zhuge, H., 2003, "Component-Based Workflow Systems Development," *Decision Support Systems*, Vol. 35, No. 4, pp. 517~536.