Information Technology Journal 13 (4): 624-633, 2014
ISSN 1812-5638 / DOI: 10.3923/itj.2014.624.633

Ontology-based Knowledge Service with Enhanced Domain Specification

Yang Tao, Zhang Tao and Zhang Yu
Key Lab of Information Network Security, Ministry of Public Security,
Third Research Institute of the Ministry of Public Security, 339 Bi Sheng Road,
Zhangjiang Hi-tech Park, 201204, Shanghai, China

Abstract: Traditional workflow research focuses on process-based workflow property, modeling technology and verification technology. The authors find that there are deficiencies in dealing with the knowledge or data representation capabilities of these process-based workflows. Although workflow can fulfill many requirements for customizing process design and execution, researchers have yet to reach a common view on how to model the dataflow conveniently and legibly, as well as how to verify the correctness of these data-sensitive work flows efficiently. In this work, a novel method is introduced to define a workflow with restricted dataflow and a verification methodology is provided to identify format, syntactic, logic and data semantic errors in this workflow. A Workflow-based Knowledge Service (WKS) system is also constructed to provide a graphic design tool and executing engine of restricted dataflow based workflow. Experiments on WKS demonstrate the capability of this method to model knowledge applications to executable workflows as well as the efficiency of its verification algorithm.

Key words: Ontology, workflow, dataflow, knowledge service, verification

INTRODUCTION

Workflow technology, which has the capability to model and execute processes in a specific business domain, is becoming a very important practical method in many research and business areas. The most important property of workflow is its capability to describe process order. Another important property is its capability to describe dataflow (Mei *et al*., 2008). Traditional workflow research has focused on the control flow facet, such as process modeling technology and process verification technology (Yu *et al*., 2012).

In recent years, researchers have expanded their works to apply workflow technology to more practical applications, such as agricultural or knowledge grid applications (Zhang, 2006). These practices have brought forward the requirements for handling the dataflow in workflow. It is difficult to use one model to map both control flow and dataflow of workflow. Hence, one solution proposed solution is to use different models to describe different flows, such as an activity or status graph in Unified Modeling Language (UML). Other proposed solutions use one model to contain both the process flow and dataflow, as dual workflow (Fan *et al*., 2007). However, these methods have convenience or legibility deficiencies.

In the agricultural domain, workflow technology can help solve problems, such as those in plant guides. Agricultural domain experts are supposed to build a plant guide workflow that allows agricultural users to utilize the workflow to acquire the knowledge they need in planting. Moreover, modeling the process of workflow requires the capability to model data representation and data transfer in topical agricultural applications.

The background of our research is an agricultural project called massive agricultural knowledge and resources management system, which aims to integrate massive data from various agricultural knowledge systems and resource databases into one portal for a more efficient use. The Agricultural Ontology Service (AOS) developed by the Food and Agriculture Organization (FAO) is used to integrate and build the relationship among these massive data. The core requirements include the following:

- Knowledge from different sub-domains (plant/animal) requires different representations
- Different users (decision-makers/technicians/ farmers) use different operation processes and
- Onespecific user may want to obtain different knowledge with different preferences (grow/gain)

To fulfill the requirements of the system, the authors introduced a novel method to define a workflow with

Corresponding Author: Yang Tao, Key Lab of Information Network Security, Ministry of Public Security, Third Research Institute of the Ministry of Public Security, 339 Bi Sheng Road, hangjiang Hi-tech Park, 201204, Shanghai, China

restricted dataflow. A single model is proposed to describe both control flow and dataflow, which maintains the convenience and legibility by restricting the dataflow hierarchy. To check the correctness of workflow, a verification methodology is also introduced, through which workflow designers can easily know the format, syntactic, logic and data semantic errors through this checker. Thus, a Workflow-based Knowledge Service (WKS) system, which provides a graphic design tool to build, verify and execute the workflow, is constructed. Experiments on the WKS show that the restricted dataflow method has the advantages of efficiency and usability compared with the traditional dataflow process method; in addition, its verification algorithm has a marked improvement in precision and efficiency.

RELATED WORKS

Traditional workflow approaches: Recent approaches to modeling dynamic behavior of process in workflow can be divided into three kinds: Unified Modeling Language (UML), Petri Nets and others.

The Unified Modeling Language (UML) is a general-purpose object modeling language, which provides capabilities to build several diagrams for process modeling. These diagrams include sequence diagrams, collaboration diagrams, state diagrams and activity diagrams. UML sequence diagrams are used to visualize specific instantiations of a UML use-case. UML collaboration diagrams visualize the flow of control through a numbering scheme in the context of classes and their static relations in structure diagrams. UML state diagrams are used to describe the lifecycle of a specific system or subsystem in a reactive manner. UML activity diagrams comprise a combination of states and Petri nets that are made more expressive to match most of the requirements of modeling the process perspective of workflows (Weske, 2012).

On the other hand, Petri Nets are based on a strong theoretical foundation and focus on properties of workflows and the application of inheritance. A Petri net, which models the control-flow dimension of a workflow is called a Workflow Net and has one source place and one sink place because any case handled by the procedure represented by the Workflow net is created when it enters the workflow management system (Van der Aalst *et al*., 2011). The case is then deleted once it iscompletely handled.

In addition to the two previous methods, other workflow modeling approaches mainly serve specific applications; for instance, the IBM MQ Workflow, is a topical commercial workflow management system supporting business process workflow modeling. Another approach is the WASA, which is a workflow prototype for scientific database application modeling.

Dataflow support in workflow: Although, the workflow modeling methods of UML and Petri Nets are mature in modeling the control process of a workflow, dealing with the dataflow remains a problem (Storrle, 2005). Some researchers have added an extension on Petri Nets called Colored Petri Nets in order to add dataflow information to workflow (Marir and Ndeta, 2013). However, this method only supports very simple dataflow check function, such as input/output verify and does not support the complex data type and data expressions.

The most recentproposal is the Dual Workflow Nets method (Trcka *et al*., 2009), which has been proposed to explicitly model the control flow and dataflow of workflow processes and can capture control/dataflow interactions. This method extends the traditional Workflow Net and combines the control flow and dataflow into a complex-value token. After giving the formal definition and its verification algorithm, the Dual Workflow Nets show its capability in performing dataflow support. The main problem of this method is its lack of experiments and supporting tools.

Workflow verification methods: Workflow verification is concerned with the logical correctness of workflow process definitions. Depending on the workflow language used, there may be different properties that have to be satisfied. For example, potential deadlocks, never-ending loops and inconsistencies are some of the potential errors of a workflow that have to be identified during verification (Meixner and Sorin, 2007).

The most traditional verification methods include UML verification, XML-scheme-based verification, Logic-based verification and Petri net-based verification (Ciesielski *et al*., 2002; Kovacs and Gonczy, 2008). The UML verification method focuses on the correctness of the diagram and interaction between diagrams, such as the use case in use case diagram and sequence diagram. The XML-scheme-based verification method focuses on the correctness of XML format of workflow files using the XML-scheme technology, such as DTD or XSL. On the other hand, logic-based verification uses formal logic to represent the workflow and then verifies the logical expression by logic check tools. Most popular methods of logic-based verification include Computational Tree Logic (CTL) and Linear Temporal Logic (LTC), both of which have mature tools, such as SPIN to verify the logic correctness (Haq *et al*., 2009). The Petri net-based verification method uses the theoretical foundation of Petri net, which defines live, bounded, strongly connected and sound properties. Using these properties and the check algorithm, the verification method can be easily done by workflow formalization.

Ontology-based verification method (Yang *et al*., 2010) is a new research topic in the field of workflow verification. The main idea for this method is to use ontology to represent the semantic of process and data and then check the connected flows to verify if their semantic contexts match.

Traditional verification methods provide control verification of the workflow, but lack dataflow verification. The new ontology-based verification method provides both overall functions; however, the workflow implement method is different from the traditional method and its efficiency is somewhat lower than the ontology search method. The agricultural project requires a new verification method to check both control flow and dataflowand must be compatible with the workflow description method we want to use.

WORKFLOW WITH RESTRICTED DATAFLOW

In a topical agricultural application, such as planting guide, the user goes through the growth periods during sowing time. In each growth period, the user can view detailed information and notes, ask questions and search the knowledge database for solution. Our restricted dataflow based workflow can represent this kind of application in the following definitions.

To represent such a knowledge application, we define the corresponding workflow as a triple tuple.

- **Definition 1:** WF=<N, F,D>, where N represents the nodes, F represents the control flows and D represents the dataflows

Node definition: Here, N is a set of nodes including:

- **Definition 2:** $N = C\cup V|S$, where C represents the Coordinator nodes, V represents the Variables nodes and S represents the service component nodes

Coordinator nodes C are used to control the executing logic and order, including {(Start), (End), (If), (Else), (EndIf), (While), (EndWhile), (Case), (When) and (EndCase)}. (Start) is the process entry and (End) is the exit. (If)/(Else)/(EndIf) and (Case)/(When)/(EndCase) are the branch flow coordinators with branch condition expressions. (While)/(EndWhile) is the loop flow coordinators with loop condition expressions.

Variable nodes V delegate the variables containing three options: define, get and set.

S refers to the service component nodescontaining built-in knowledge retrieval, personality management and external services. A detailed definition of S is presented below.

S is a set of service components that processes the main operations of the workflow. Its types include:

- Html output component, which indicates when, how and what to show on the web browser
- Knowledge retrieval component, which is established to retrieve knowledge from the distributed massive knowledge database
- Personality management component, which supports the user personality catch and use as well as provides the set/get interface
- External knowledge service. In addition to the previous three build-in service components, external services are often required as supplements to accomplish some functions (for instance, query weather, crop price)

These service components are to be called when the applications are executed. The S is defined as:

- **Definition 3:** S = <U.T1/O>, where U is the service URL indicating how to call the service, T is the service type as mentioned above and I/O representthe input and output interfaces of the service functions, respectively

Control flow definition: F represents a set of control flows defined as:

- **Definition 4:** $F|N\times N$, where N represents the nodes

The most important property of control flow is connectivity. Hence, for each two nodes of workflow beside V, there must be a control path connected with two nodes. The isolated node is not allowed except in variables nodes. We then define each element of N to have multiple entries and one exit control flow property except the (If), (Case) and (End) coordinators in our workflow model. It can be concluded as:

- **Definition 5:** Multi entries and one exit

$\forall$ n$\in$N-{ (If), (Case), (End)}, there is no more than one $n_i\in Nn\times n_i\in F$, where N represents the nodes and F represents the control flows.

Dataflow definition: The last part of WF is D, which means a set of dataflows. Although, dataflow also links two nodes, its properties vary hugely from the control flow. First, dataflows only occur in the I/O of services and variables. The detailed definition of D is:

- **Definition 6:** $D\subseteq(S.I\times S.O)\cup(V\times S.O)$, where S represents the service components and V refers to the variable nodes pre-defined as ontology instances

Next, we define each element of N that can have dataflow to have both multiple entries and exits dataflow property. The last dataflow property is the data type match rule, which means either side of the dataflow must have the same data type.

We must point out that this definition restricts the hierarchy of dataflow to a single level. Given that there is no isolated node in the control flow except for V and dataflow does not include the definition of (V×V), it means that the max length of dataflow link without orthogonal control flow is 1. Figure 1 shows two different cases, the left one is correct and the right one has an error due to its unconnected node "S: Output". This feature gives us the ability to handle the most convenient and efficient dataflow for the verification algorithm design.

Workflow operation: We then provide definitions for WF operations:

- **Definition 7:** WF operations:

Pre(n) = $\{\forall(n, n_i)\in F\}$ indicates all control flows with the prefix n.

PreD(n) = $\{\forall(n, n_i)\in D\}$ indicates all data flows with the prefix n.

Suf(n) = $\{\forall(n_i, n)\in F\}$ indicates all control flows with the suffix n.

SufD(n) = $\{\forall(n_i, n)\in F\}$ indicates all data flows with the suffix n.

Des(n) = $\{n_i\in N|(n, n_i)\in F\}$ indicates the direct descendant of N.

Asc(n) = $\{n_i\in N|(n_i, n)\in F\}$ indicates the direct ascendant of N.

AD(n) = Des(n)$\cup\{n_i\in N|n_j\in AD(n), (n_j, n_i)\in F\}$ indicates all the descendants of N.

AA(n) = Asc(n) $\cup$ $\{n_i\in N|n_j\in AD(n), (n_i, n_j)\in F\}$ indicates all the ascendants of N.

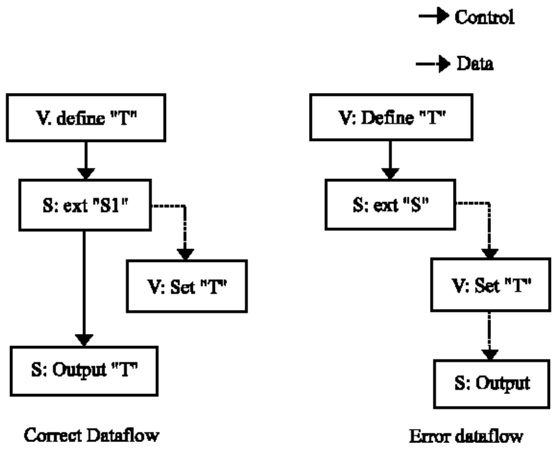


Fig. 1: Restricted dataflow

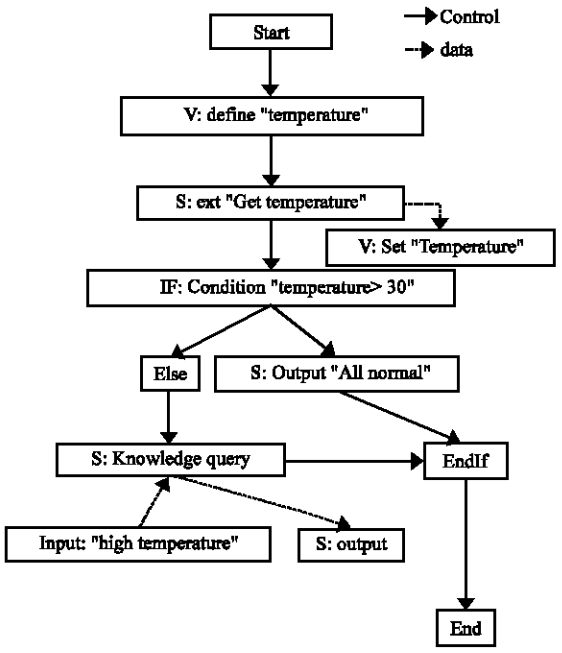


Fig. 2: Workflow example

These operations can be used in the following error definition and verification algorithm of the workflow.

Example: We provide a simple example of this workflow with restricted dataflow (Fig. 2). The solid line refers to the control flow and the dotted line refers to the dataflow. The process of this sample involves the branch process on different temperatures. The workflow first defines the variable temperature and then calls an external service to obtain the temperature and send the value to the variable. After being determined using the temperature range, the workflow outputs the "all normal" messages or shows the knowledge on high temperature planting guide.

WORKFLOW VERIFICATION

Correctness of workflow: To verify the workflow, we must first define the correctness of workflow. To define the correctness, we need to identify the error. In this workflow, errors are defined as follows:

- **Definition 8.** Format error

Format error refers to the error of workflow file. It also includes the following states:

- $|Suf((Start))| >= 1$, which means (Start) is not the first node
- $|Pre((End))| >=1$, which means (End) is not the last node
- $|Suf((If))| != |Pre((EndIf))|$, which means that there is a mismatch on (If)
- $|Suf((Case))| != |Suf((EndCase))|$, which means that there is a mismatch on (Case)
- $|Suf((While))| != |Suf((EndWhile))|$, which means that there is a mismatch on (While)

- **Definition 9:** Syntactic error

Syntactic error refers ti the flow error of the workflow. It includes the following states:

$\exists n \in N-\{(If), (Case), (End)\}, |Pre(n)|>1$, which refers to multiple exits.

$(AD((Start)) != N) \in (AA((End)) != N)$, which means that there are isolated flows or the node does not connect with any flow.

$\exists n_i, n_j \in N-\{ (If), (Case), (End)\} \mid n_i \in AA (n_j) \in n_j \in A (n_i)$, refers to the latency deadlock.

- **Definition 10:** Logical error

Logical error refers to errors identified on the logic level. This error can often take place in the actual execution. Logical errors include condition expression error, inaccessible path on branch condition, latency deadlock on branch/loop condition and unset variable access.

In addition to the normal process logical error in the workflow, this restricted dataflow-based workflow focuses more on dataflow errors. The logical error of process flow mainly contains an inaccessible path which means some path of workflow can never be visited and latency deadlock, which means some loop can never stop under specified conditions. The logical error of dataflow mainly focuses on the unset variable access error, which means a variable access action cannot access the required data or the variable cannot initialize correctly. We must point out that this workflow uses restricted dataflow because such dataflow can fulfill most of the function requirements we meet; in addition, the design tool and the logical error check algorithm are moreefficient and usable than the unrestricted one.

By defining these workflow errors, we can give the correctness of workflow as follows:

- **Definition 11:** Correct workflow

A correct workflow is a workflow that does not contain any format, syntactic and logical errors.

Verification algorithm: We also utilize our algorithm to verify the workflow. Most of the old methods follow the same path: for each error condition, the workflow is checked to determine whether it occurs, or a simple format checker such as DTD/XSL is used. For our definition, there are various errors in the workflow and some errors only take place during specific steps. We give a multi-dimension verification method, which contains four steps.

Step 1: XML-based scheme check

In this step, we use a pre-defined XSL checker to check the node definition and relation. It can also filter some format errors. Part of our checker is shown below:

```
<xs:element name="WorkFlow">
<xs:complexType>
<xs:sequence>
<xs:element name="variables">
<xs:complexType>
<xs:sequence>
<xs:element name="variable">
<xs:complexType>
<xs:attribute name="name" />
<xs:attribute name="type" />
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="sequences">
<xs:complexType>
<xs:sequence>
<xs:element name="sequence" />
<xs:element name="while">
<xs:complexType>
<xs:simpleContent>
<xs:extension base="xs:string">
<xs:attributename="condition" />
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
...
```

Step 2: Workflow parameter verification

In this step, we checked the variable defined and used states, IF/Case/While condition expressions, validity of the external service and output format. The main goal for this step is to ensure that the following check steps runwith no parameter errors.

Step 3: Logic check

In this step, we begin the main verification operation of our algorithm; here, all flow errors and some logical errors should be found in this step except for the inaccessible path and latency deadlock. These are to be solved in the step automatic execution.
The main algorithm is as follows:

```
Algorithm 1: Optimized model verification with dataflow(OMV)
Input: XML Structure M of workflow
Return: isCorrect refers M is logic correctness and other results contain the error nodes.
0 Place all nodes of the M into collection N, all control flows into collection F, all data flows into collection D,then create four empty collections N1, N2,N3, and NErr and a flag stack NStack.
1 add (start) to N1, N3;
2 While (true)
3 begin
4 Remove all elements in N1 from N;
5 for each element n in N1 do
6 if typeof(n)∈C-{ (If), (Case), (End)}
7 for each flow f has f.startnode=n in F do
8 begin
9 if checkflow(f) not is true
10 begin
11 add f to Nerr;
12 if f.endnode not in N3
13 add f.endnode to N3
14 end;
15 add f.endnode to N2
16 end;
17 elseif n is the begin tag of (If), (Case), (End)
18 add (n, IFFlag/CaseFlag/WhileFlag) to NStack;
19 elseif n is the end tag of (If), (Case), (End)
20 remove top IFFlag /CaseFlag /WhileFlag from NStack;
21 elseif n∈S∪V
22 foreach flow f has f.startnode=n or f.endnode=n in PreD(n) or SufD(n) do
23 if checkdataflow(f) not is true
24 put f to Nerr;
25 elseif typeof(n) is (end)
26 break;
27 clear N1;
28 move all element from N2 to N1;
29 end;
30 if N, Nerr, Nstack not empty
31 isCorrect=false;
32 else
33 isCorrect=true;
34 return isCorrect,N,Nerr,Nstack to caller;
```

The main part of our verification algorithmis the logic check algorithm. It starts from the Start nodes, literally checking whether or not all workflow paths from this node are correct. When control nodes, such as (If)/(Case)/(While) are encountered, the algorithm places the start flag to the stack and waits for the nearest (EndIf)/(EndCase)/(EndWhile) to clear this flag. When nodes with dataflow are encountered, the algorithm also checks the logical correctness of the data flows using ontology reasoning. Through these steps, the algorithm can find most of the workflow logical errors and data errors in the restricted dataflow.

Step 4: Automatic execution

In this step, the execution engine checks the branch point of workflow, generates related variable and service list using branch testing case generator, simulates the execution the workflow and then marks the node cover status. After finishing the execution of the testing case, the latency deadlock and inaccessible branch are also found in this step.

EXPERIMENT AND EVALUATION

Experiment: The research topic belongs to an agricultural project called the massive agriculture knowledge and resource management system. The main purpose of this project is to integrate massive data from various agricultural knowledge systems and resource databases into one portal to make it more user-friendly. Using the system, domain experts can build some customized agriculture applications flexibly and each kind of user uses these applications to acquire the knowledge they need.

For an experiment using this approach and an infrastructure of the agricultural project, the authors provided the design tool and executing engine, which we call the Workflow-based Knowledge Service system or WKS. Using this system, the authors built several actual agricultural applications, including a cucumber planting guide and a decision-maker on regional crop selection. The verification method was also applied to these applications in order to check its capability and efficiency. The details of the experiments are presented in the next section.

The authors of the current work developed WKS by Eclipse 3.4 Galileo using Java 1.5 and MySQL 5.0. The design tool was built using a GMF framework and then distributed in Eclipse RCP package. The executing engine was a Java web application hosted by Tomcat 6.0. Web and the database server was conducted on an Intel Pentium 4 1.8 GHz PC with 2 GB RAM, running Windows Server 2003. The client was conducted on Intel Centrino Duo T2400 1.83 GHz PC with 2 GB RAM, running Windows XP sp3, which ran the design tools or web browser.

Domain experts used the design tools to design agricultural applications and then used the system release function to deploy the application to the executing engine (Fig. 3).

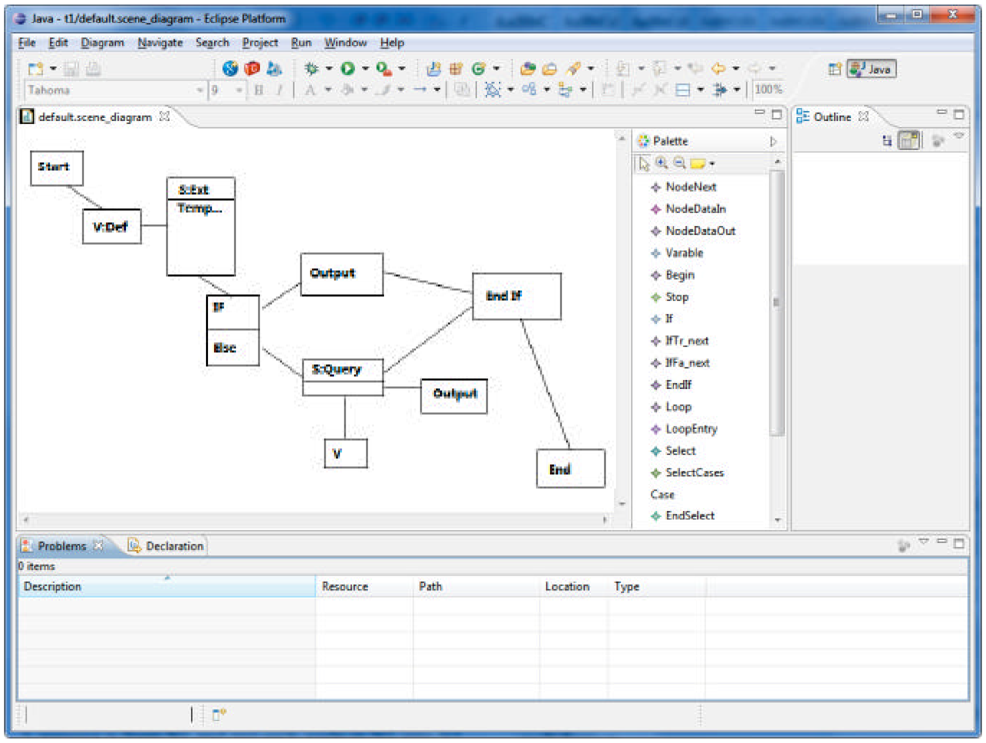
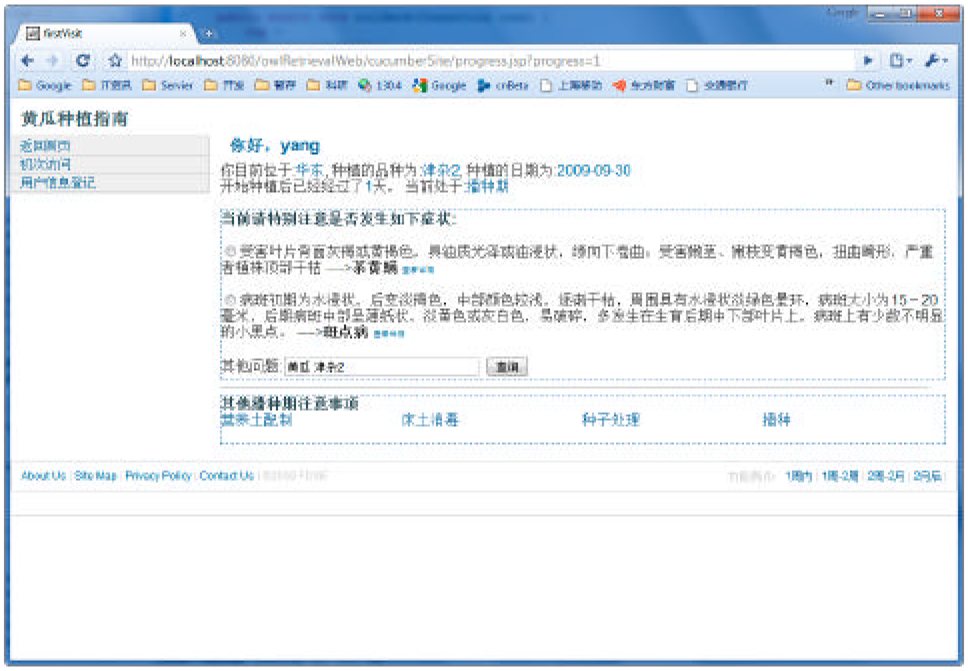


Fig. 3: WKS design tool


http://localhost:8080/owlRetrievalWeb/cucumberSite/progress.jsp?progress=1

黄瓜种植指南

返回首页
初次访问
用户信息登记

你好，yang

你目前位于 华东, 种植的品种为 津杂2, 种植的日期为 2009-09-30
开始种植后已经经过了1天。 当前处于 播种期

当前请特别注意是否发生如下症状:

◎ 受害叶片背面灰褐或黄褐色，具油质光泽或油浸状，叶缘向下卷曲，受害嫩茎、嫩枝变黄褐色，扭曲畸形，严重者植株顶部干枯 —>茶黄螨

◎ 病斑初期为水浸状，后变淡褐色，中部颜色较浅，逐渐干枯，周围具有水浸状淡绿色晕环，病斑大小为15－20毫米，后期病斑中部呈薄纸状，淡黄色或灰白色，易破碎，多发生在生育后期中下部叶片上，病斑上有少数不明显的小黑点。 —>斑点病

其他问题: 黄瓜 津杂2 查询

其他播种期注意事项
营养土配制 床土消毒 种子处理 播种

About Us | Site Map | Privacy Policy | Contact Us | 1周内 | 1周-2周 | 2周-2月 | 2月后

Fig. 4: WKS executing result

After completing the application, farmers or other users can access the application using a web browser (Fig. 4). The executing engine was designed for the use of an actual Chinese farmer and the web page on Fig. 4 shows the growth stage and possible diseases from a cucumber planting guide. All of the visible and invisible contents on this web page are related to the HTML output nodes in the workflow file. The region R1 lists in

Table 1: Properties on correctness of workflow files

Error type	Error count in draft one (759 total lines)	Error count in manual checked (781 total lines)
Format error	12	0
Syntactic error	26	0
Logical errors		
Inaccessible path	5	0
-Deadlock	8	0
-Dataflow error	19	0
-Others	11	0
Total	71	0

Table 2: Verification results (FE: Found error, UF: Unfound error, IF: Incorrect found)

	Error found in draft one			Error found in manual checked		
Error type	FE	UF	IF	FE	UF	IF
Format error	12	0	0	0	0	0
Syntactic error	24	2	0	0	0	0
Logical errors						
-Inaccessible path	5	0	1	0	0	0
-Deadlock	8	0	0	0	0	0
-Dataflow error	18	1	2	0	0	1
-Others	9	2	0	0	0	1
Total	66	5	3	0	0	2

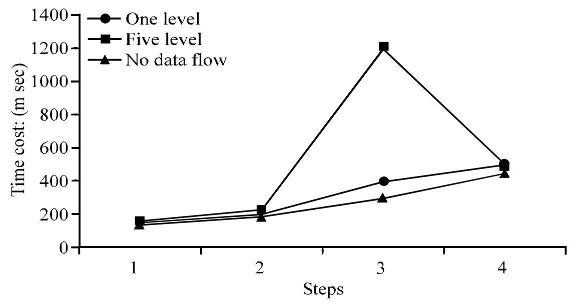


Fig. 5: WKS running time analysis

detail the basic information of the workflow user, which wasgenerated from the environment variables, user personalities and external services by the workflow file. On the other hand, region R2 lists possible diseases generated from the agricultural knowledge retrieval components call by the executing engine.

Capability of verification algorithm: In order to check the capability of the verification method, the authors used two workflow application files: a draft workflow of cucumber planting guide, which contains several kinds of errors and a manually checked version of previous one, which contains no error. The authors used the verification method step by step to check the workflow error. The properties of these two workflows are listed in Table 1. The verification result can be found in Table 2.

When processing the first draft workflow, the verification algorithm found 93% errors, with 100% format errors, 92% syntactic errors and 91% logical errors. In particular, it found 95% dataflow errors, which the other workflow verification method could not provide. It also found three incorrect results, containing a path check question and two dataflow questions. Most of these errors were caused by the data type mismatch when conducting the branch test and dataflow check. When processing the next manual checked workflow, the verification algorithm found two incorrect results, containing a dataflow question and an expression check question.

From these two experiments, we found that our algorithm verified the workflow with restricted dataflow with an acceptable precision and recall rate.

Efficiency: We compared the efficiency of the method with a different workflow definition with no limited dataflow hierarchy. The main difference can be found in Step 3. This algorithm does not need to conduct a recursion search in line 22 for dataflow validation. In addition, when faced with no limited dataflow hierarchy, a recursion algorithm must be used to find the entire mistake data mapping.

Another method, which was compared withthe traditional no dataflow verification. The greatest difference between these two methods was whether to perform the work in line 23.

The details of the running time analysis can be found in Fig. 5. The first result with the round node is the time cost of the method. The second result with square nodes refers to the time cost of the multi-hierarchy (as a test, we changed the dataflow to five levels in the workflow) dataflow check method and the last result with triangle nodes is the time cost of the no dataflow check. From the experiment, it can be concluded that the workflow with restricted dataflow lost little efficiency with no dataflow check, but a non-restricted one may lose much more.

CONCLUSION AND FUTURE WORK

In this study, the authors introduced a novel method to define a workflow with restricted dataflow.It utilizes a single model to describe the control flow and dataflow. It is also convenient and legible because it restricts the dataflow hierarchy.

We found this method to be efficient in modeling and executing agricultural applications. First, the method is a workflow-based logic control system in a functional perspective, which supports different operations of a service. The process support of the workflow contains an implied sequence process, (If) (Case) for branch process and (While) for loop process. It does not support the parallel process because it nearly never occurred in this project. It also has a scene-based knowledge representation that uses the html output node to support the different representation of knowledge. The workflow definition supports the internal and external services to perform the knowledge operation of the system. Next, to ensure operational capability, we developed the design tool and executing engine to support this workflow. From its practical use, the workflow has high operational capability and the restricted dataflow method is also found to be efficient. Finally, considering the flexibility, the work flow supports the condition expression and provides the definition and set/get functions of the variables, as well as supports the external services of the system.

From these properties of the workflow with constricted dataflow, we conclude that this workflow method fulfilled the project requirements of massive agriculture knowledge and resources management system. However, this mechanism still has some problems.The most important problem of this approach is its capability to support multiple hierarchy dataflow. Through actual usage, most domain users view the disadvantage as a big problem. Another problem is the automatic error modification proposal. This is a difficult problem in workflow verification and validation. The last problem is how to make the design tool user-friendly as well as to create an execute engine with a more user-friendly access method. Currently, it is a big problem for training domain experts who are learning the system. End users also encounter confusions on the operation of a generated web page when executing the workflow.

In our future research, we intend to focus on the following topics:

- Automatic error modification after verification and validation and
- A more efficient verification and validation algorithm

ACKNOWLEDGMENTS

This study is supported by the National Key Technology Research and Development Program of the Ministry of Science and Technology of China (2012BAH95F03).

REFERENCES

Ciesielski, M.J., P. Kalla, Z. Zheng, and B. Rouzeyre, 2002. Taylor expansion diagrams: A compact, canonical representation with applications to symbolic verification. Proceedings of the Design, Automation and Test in Europe Conference and Exhibition, March 4-8, 2002, Paris, pp: 285-289.

Fan, S., W. Dou and J. Chen, 2007. Dual workflow nets: Mixed control/data-flow representation for workflow modeling and verification. Proceedings of the International Workshops Advances in Web and Network Technologies and Information Management, Volume 4537, June 16-18, 2007, Huang Shan, China, pp: 433-444.

Haq, I.U., A. Paschke, E. Schikuta and H. Boley, 2009. Rule-based workflow validation of hierarchical service level agreements. Proceedings of the Workshops at the Grid and Pervasive Computing Conference, May 4-8, 2009, Geneva, pp: 96-103.

Kovacs, M. and L. Gonczy, 2008. Simulation and formal analysis of workflow models. Electr. Notes Theor. Comput. Sci., 211: 221-230.

Marir, F. and J. Ndeta, 2013. Knowledge enhanced framework for designing e-workflow systems. Proceedings of the 5th International Conference on Advances in Databases, Knowledge and Data Applications, January 27-February 1, 2013, Seville, Spain, pp: 143-149.

Mei, L., W.K. Chan and T.H. Tse, 2008. Data flow testing of service-oriented workflow applications. Proceedings of the ACM/IEEE 30th International Conference on Software Engineering, May 10-18, 2008, Leipzig, pp: 371-380.

Meixner, A. and D.J. Sorin, 2007. Error detection using dynamic dataflow verification. Proceedings of the 16th International Conference on Parallel Architecture and Compilation Techniques, September 15-19, 2007, Brasov, pp: 104-118.

Storrle, H., 2005. Semantics and verification of data flow in UML 2.0 activities. Electr. Notes Theoretical Comput. Sci., 127: 35-52.

Trcka, N., W.M.P. van der Aalst and N. Sidorova, 2009. Data-flow anti-patterns: Discovering data-flow errors in workflows. Proceedings of the 21st International Conference on Advanced Information Systems Engineering, June 8-12, 2009, Amsterdam, The Netherlands, pp: 425-439.

Van der Aalst, W.M.P., K.M. Van Hee, A.H.M. ter Hofstede, N. Sidorova, H.M.W. Verbeek, M. Voorhoeve and M.T. Wynn, 2011. Soundness of workflow nets: Classification, decidability and analysis. Formal Aspects Comput., 23: 333-363.

Weske, M., 2012. Business Process Management: Concepts, Languages, Architectures. 2nd Edn., Springer Press, USA., ISBN: 978-3-642-28616-2, Pages: 403.

Yang, T., Y. Wu and W. Zhao, 2010. An ontology-based agriculture knowledge service workflow. Proceedings of the Web-Based Education, March 15-17, 2010, Sharm El Sheikh, Egypt, pp: 16..

Yu, J., Q.Z. Sheng, J. Han, Y. Wu and C. Liu, 2012. A semantically enhanced service repository for user-centric service discovery and management. Data Knowl. Eng., 72: 202-218.

Zhang, J., 2006. Ontology-driven composition and validation of scientific grid workflows in Kepler: A case study of hyperspectral image processing. Proceedings of the 5th International Conference on Grid and Cooperative Computing Workshops, October 2006, Hunan, pp: 282-289.