Paper:

Integrating Process and Work Breakdown Structure with Design Structure Matrix

Jonathan Lee*, Whan-Yo Deng*, Wen-Tin Lee*, Shin-Jie Lee*
Kuo-Hsun Hsu, and Shang-Pin Ma*****

*Department of Computer Science and Information Engineering, National Central University, Jhongli, Taiwan
E-mail: {yjlee, jass, wtlee, jielee}@selab.csie.ncu.edu.tw
**Department of Computer and Information Science, National Taichung University, Taichung, Taiwan
E-mail: glenn@mail.ntcu.edu.tw
***Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung, Taiwan
E-mail: albert@mail.ntou.edu.tw
[Received November 3, 2009; accepted May 20, 2010]

In software development, project plans document scope, cost, effort, and schedule, guide project managers, and control project execution. Developing a project plan without incorporating how an organization doing things – i.e., organizational culture – may lead to project failure. To ensure stable process performance and to benefit from organizational culture, it is crucial that organizational processes be taken into account in project planning. Organizational processes enable stable process performance across an organization and provide a basis for cumulative, long-term benefits to the organization. In proposing a systematic approach that supports bi-directional transformation between processes and the Work Breakdown Structure (WBS), we propose Process2WBS and WBS2Process to assist project managers in project planning with an organization's set of standard processes. Process2WBS consumes processes and transforms them into a WBS with Design Structure Matrix (DSM) analysis, and WBS2Process transforms the WBS with project-specific information into executable processes expressed in XPDL.

Keywords: process management, project management, project planning, design structure matrix

1. Introduction

The Work Breakdown Structure (WBS) is a hierarchical list of project tasks that defines the scope of a project, which translates into effort, timeline, and budget. Taking the time to map out the WBS saves significant effort in project execution by helping avoid rework and false starts [1–3]. An important WBS planning objective is project scheduling. Although considerable research [4] has been focused on project scheduling, little work has accounted for organizational processes in the project planning phase. An organization's set of standard processes provides project managers with knowledge sharing and lessons learned. Developing a project plan without incorporating how an organization does things, namely, organizational culture, may cause a project to fail. To ensure stable process performance and to benefit from organizational culture, it is crucial that organizational processes be taken into account in project planning. Organizational processes enable stable process performance across the organization and provide a basis for cumulative project development experience. Continuous improvement of organizational processes also provides long-term benefits to the organization.

A process is a set of activities connected to control nodes providing decision support and flow logic. Dependence among activities is complex in a project process. Managing complex dependence among activities is thus a competency required for successful process execution. Conventional process management tools provide process representation graphically, however, not allowing for common feedback and cyclic activity dependence. The Design Structure Matrix (DSM) devised by D.V. Steward [5] serves as system analysis for representing processes and their relationships in a square matrix and for analyzing feedback and cyclic process interaction.

The DSM is a square matrix with identical row and column labels to identify dependence between tasks and to sequence the engineering design process. This complexity management tool designs and optimizes a complex system, project tasks, and organization structure. T. R. Browning [6] reviewed four DSM applications to demonstrate their usefulness in product and process development, project planning and management, system engineering and organization design. The four DSM applications, which include component-based, team-based, activity-based, and parameter-based DSM, are categorized into Static DSM and Time-based DSM. The DSM uses several types of analysis to optimize a complex system and project tasks, such as partitioning, clustering, and simulation [7, 8].

Improving process execution efficiency and process

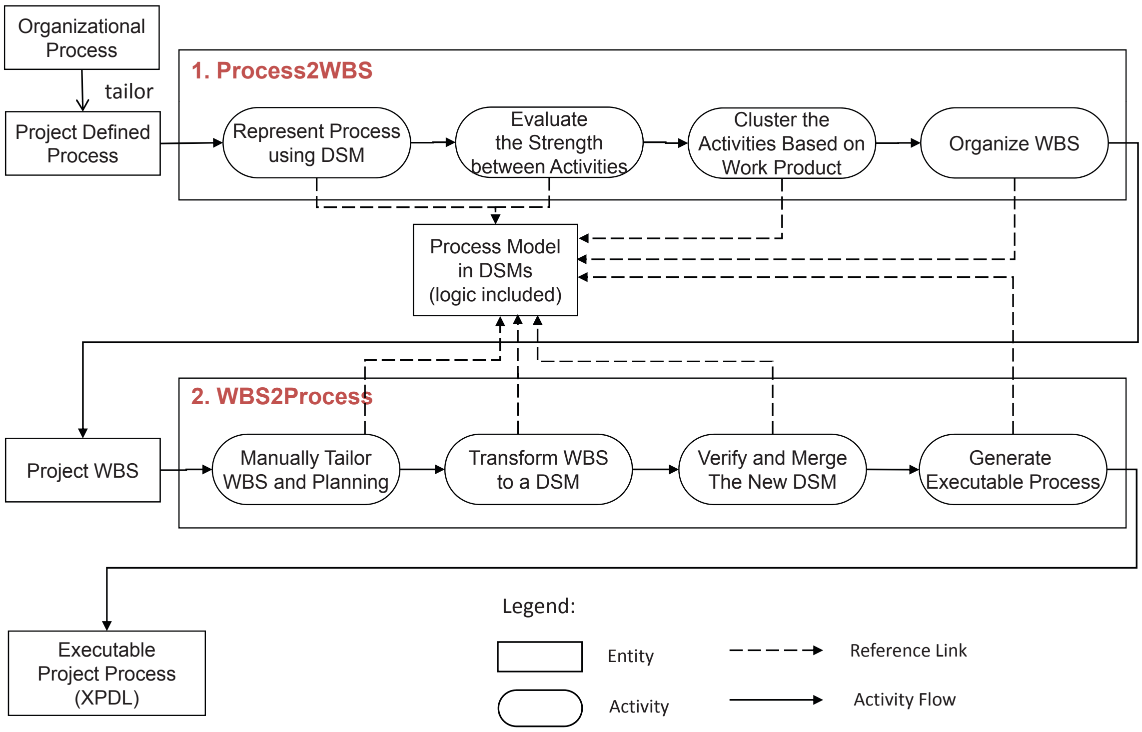


Fig. 1. Overview of our approach.

control requires a workflow engine to execute the project process automatically. A project process is further enhanced using a process definition language such as XML Process Definition Language (XPDL), a *de facto* standard promoted by the Workflow Management Coalition (WfMC) [9]. XPDL is an open flexible process definition standard enabling process designers to define project processes and extension attributes, and a process definition language managed by a workflow engine.

As a continuation of previous work on requirements engineering [10–15], we propose a systematic approach supporting bi-directional transformation between processes and a work breakdown structure – Process2WBS and WBS2Process – to assist project managers in project planning with an organization's set of standard processes.

- Process2WBS consumes processes and transforms them into a WBS. A WBS template derived from a project-defined process, increases WBS conformity with the project-defined process. The domain-mapping table, mapped from a process to the DSM and from the DSM to the WBS, helps calibrate mapping relationships between a process and a WBS. A clustering algorithm is developed to analyze the degree of strength among activities to group activities based on deliverables.

- WBS2Process transforms a WBS with project-specific information into executable processes expressed in XPDL format. The DSM maintains processes, subflows, and activities or tasks in a WBS based on WBS editing constraints. The DSM and the original DSM produced by Process2WBS are merged by synchronizing activities, input logic, and output logic. WBS2Process then translates the merged DSM into an XPDL file by mapping from the DSM to XPDL format. An XPDL file also documents project-specific information in corresponding tags.

This paper is organized as follows: Section 2 discuss in depth how to integrate processes and WBS with the DSM. Section 3 shows an example demonstrating our proposed approach. Section 4 reviews related work, and Section 5 presents conclusions.

2. Integrating Process and WBS

Discussing how to incorporate an organization's set of standard processes with the WBS and how to transform the WBS into an executable process involve the two main features shown in **Fig. 1**.

- Transform Process to the WBS (Process2WBS): When a project is initiated, project managers may set up project-defined processes by processes tailored from organizational processes based on tailoring criteria and guidelines. A project-defined process provides a basis for planning and conducting the project's tasks and activities. The WBS defines and groups a project's tasks or work elements to help project managers organize and define the project's total work scope, so the project-defined process must be transformed into a WBS in the initial phase of project planning. **Fig. 1** "Process2WBS" consumes the project-defined process and generates the corresponding WBS. Here we use Microsoft Project as our WBS tool to show the transformation between processes and the WBS. During Process2WBS transformation, processes are represented in the DSM, and dependence is analyzed by a clustering algorithm in the DSM. It is crucial that the DSM describe feedback and cyclic task dependence since most engineering applications exhibit such cyclic behavior.

- Transform the WBS to Process (WBS2Process): After transforming the WBS from the project-defined process, project managers may edit the WBS for task assignment, cost estimation, predecessor constraints, and scheduling. Improving process execution efficiency and better process control requires a workflow engine to execute the project process automatically. The WBS is useful for project cost estimation and project control, but clumsy in supporting automatic process execution, so a WBS with project-specific information must be transformed into an executable process. **Fig. 1** shows the WBS and generates the executable process in XPDL format. Because support of activity dependence logic differs between XPDL and Microsoft Project, process logic of the project-defined process is maintained in the DSM during WBS2Process processing.

2.1. Process2WBS

The purpose of Process2WBS is to incorporate the benefits of an organization's set of standard processes in the project WBS. The project-defined process is tailored from the organization's set of standard processes based on the tailoring criteria and guidelines with basis activities or tasks to execute a project, so project managers can use a project WBS template containing basic activities and tasks transformed from the project-defined process to develop the WBS during project planning.

2.1.1. Representing the Process Using the DSM

Step 1 of Process2WBS is to represent the process using the DSM. The activity-based DSM captures activities and their information flow. **Fig. 2** maps how the DSM models process concepts.

Our approach models major entities in the XPDL schema definition as process concepts in the DSM. The *Package* acts as a container for grouping individual process definitions and associated entity data applicable to all process definitions and also has a number of common attributes for the process definition entity (author, version, etc.). Since an XPDL file contains only one package, the *Package* is modeled as an activity-based DSM, including multiple processes.

The XPDL includes five activity types. To distinguish these in process concept, activity types are modeled as an element in an activity-based DSM with the extension attribute "ActivityType." *Participant/Application* describes

Process Concept	cardinality	DSM Concept
Package	1:1	DSM
Process	n:1	DSM
Activity (SubFlow/ Task/ BlockActivity/ Route/ Event)	1:1	Activity and it's Extension Attribute: Activity Type
Participant/ Application	1:1	Extension Attribute: Performer
Artifact	1:1	Extension Attribute: Input Artifact/ Output Artifact
Transition	1:1	Information Flow
Swimlane (Pool/ Lane)		None
Message Flow		None

Fig. 2. Domain concept mapping between process and DSM.

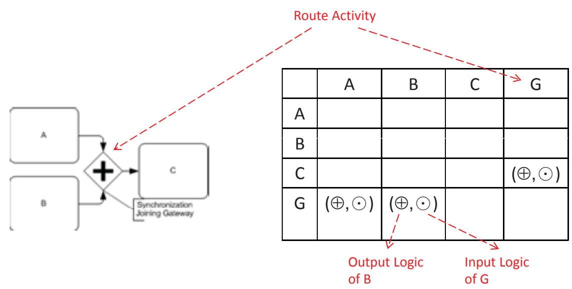


Fig. 3. Route activity in an activity-based DSM.

resources acting as the performer of activities in the process definition. This may be useful in assigning tasks to resources when editing the WBS.

We capture the *Participant/Application* as an extension attribute of an activity, which in turn captures the *Artifact* in the process concept for the same reason. The *Transition* in the process describes possible transitions between activities and conditions enabling or disabling them – transitions – during execution. An activity-based DSM models the *Transition/Information* flow as an $n \times n$ square matrix. *Swimlane* facilitates the graphical layout of a collection of processes and may designate participant information at the process level. *Swimlane* is not used during transformation between the process and the WBS, and is thus omitted from the DSM. *Message Flow* is described by the message coordination among *Swimlanes*, and is omitted from the DSM for the same reason as *Swimlane*.

The *Route Activity* uses transition restrictions (activity subelements) to implement complex routing logic, e.g., combining XOR and AND split conditions on outgoing transitions from an activity and combining XOR and AND join conditions on incoming transitions to an activity. The *Route Activity* is a "dummy" activity enabling "cascading" transition conditions to be expressed, e.g., of the type "IF Condition1 THEN DO Activity1 ELSE IF Condition2 THEN DO Activity2 ELSE DO Activity3 ENDIF" in a process. The DSM cannot deal with the above issue if the route activity is omitted.

Figure 3 shows the workflow pattern "Synchronization" and its corresponding DSM representing a Gateway as an activity. The "Synchronization" workflow pattern includes three activities and a JOIN gateway. The corre-

Score	Explanation
0	There are no transitions between activities.
1	There is more than one transition between activities.
2	There is more than one transition between activities. The target activity requires the output artifact of source activity.
3	There is more than one transition between two activities and the source activity cooperates with the target activity to develop the output artifact of the source activity.

Fig. 4. Relationships between activities.

sponding DSM has four activities – A, B, C, and G, where G indicates the JOIN Gateway – and 2-tuples represent information flows: (output logic of source activity, input logic of target activity). As symbols of the information flow, $\oplus$ is AND, $\odot$ denotes OR, and XOR is represented as $\otimes$.

2.1.2. Evaluating Strength Between Activities

After representing the project-defined process using the DSM, relationships among activities are evaluated to establish the WBS by grouping relevant activities or tasks.

Figure 4 shows scores of relationships, classified into four degrees by scoring from 0 to 3 to express strength between activities. If no transitions exist between two activities, then the score between them is 0. If more than one transition exists between them, the score is 1. If more than one transition exists between two activities and the target activity requires the output artifact of the source activity, the score is 2. If more than one transition exists between two activities and the source activity cooperates with the target activity to develop the output artifact of the source activity, then the score is 3. The DSM, called a "strength DSM," is then evaluated based on defined scores in **Fig. 4**.

2.1.3. Clustering Activities Based on Work Products

Clustering activities based on work products groups activities or tasks based on work products, so major activities producing work products are required as input for this step. Other required input is the DSM with evaluated scores generated in the previous step offering strength relations for each pair of activities. The goal of clustering is to group interrelated activities into a cluster based on the strength between activities. The clustering algorithm is divided into three steps:

1. Normalizing the DSM
2. Obtaining the strength DSM for each activity pair
3. Clustering based on major activities and strength DSM

The initial step of clustering is to normalize the DSM. Normalizing is making the strength relation of each pair of activities between 0 and 1. The transitive relation applies to deriving strength for each pair of activities. If the

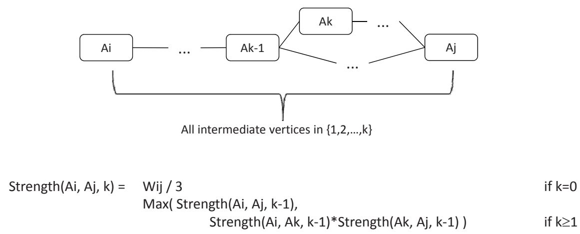


Fig. 5. Obtaining weighting scores between activities.

```
For k = 1 To n
    For i = 1 To n
        For j = 1 To n
            If i = j Then
                Strength( i, j) = 1
            Else
                If (Strength( i, j) < Strength( i, k) * Strength(k, j)) Then
                    Strength( i, j) = Strength( i, k) * Strength(k, j)
                End If
            End If
        Next j
    Next i
Next k
```

Fig. 6. Algorithm for strength derivation.

		1	2	3	4	5	6	7	8	9	...	26
Assign_Project_Manager	1	0	0	0	0	0	0	0	0	0		0
Review_Proposal	2	0	0	0	3	0	3	0	0	0		0
Is_Proposal_Approved?	3	0	3	0	0	0	0	0	0	0		0
Revise Proposal	4	0	0	3	0	0	0	0	0	0		0
Judge_Project_Type	5	1	0	0	0	0	0	0	0	0		0
Edit_Proposal	6	0	0	0	0	1	0	0	0	0		0
Submit_Proposal	7	0	0	3	0	0	0	0	0	0		0
Develop_PIP	8	0	0	0	0	0	0	0	0	1		0
Is_Project_Accepted?	9	0	0	0	0	0	0	1	0	0		0
…	…											
PPQAP	26	0	0	0	0	0	0	0	2	0		0

(a) Strength DSM

		1	2	3	4	5	6	7	8	9	...	26
Assign_Project_Manager	1	1	0.111	0.111	0.111	0.333	0.111	0.111	0.074	0.037	…	0.049
Review_Proposal	2	0.11	1	1	1	0.333	1	1	0.444	0.333	...	0.296
Is_Proposal_Approved?	3	0.11	1	1	1	0.333	1	1	0.444	0.333	...	0.296
Revise Proposal	4	0.11	1	1	1	0.333	1	1	0.444	0.333	…	0.296
Judge_Project_Type	5	0.33	0.333	0.333	0.333	1	0.333	0.333	0.222	0.111	...	0.148
Edit_Proposal	6	0.11	1	1	1	0.333	1	1	0.444	0.333	...	0.296
Submit_Proposal	7	0.11	1	1	1	0.333	1	1	0.444	0.333	...	0.296
Develop_PIP	8	0.07	0.444	0.444	0.444	0.222	0.444	0.444	1	0.333	...	0.666
Is_Project_Accepted?	9	0.04	0.333	0.333	0.333	0.111	0.333	0.333	0.333	1	...	0.222
…	…	...	...	...	...	...	...	...	..		1	…
PPQAP	26	0.05	0.296	0.296	0.296	0.148	0.296	0.296	0.667	0.222	…	1

(b) DSM after Strength Derivation

Fig. 7. DSM after strength derivation.

strength between A and B is 0.5 and the strength between B and C is 0.5, we derive the strength between A and C as 0.5*0.5=0.25. The strength between A_i and A_j (**Fig. 5**) is Strength(A_i, A_j, k) and there are k nodes in the path from A_i to A_j.

If there is a direct relationship from A_i to A_j, we define the strength as $W_{ij}/3$, where W_{ij} is the evaluated strength between A_i and A_j. There are two candidate paths from A_i to A_j: either one only using nodes in set $\{1, \ldots, k\}$ or one going from i to $k+1$ and from $k+1$ to j.

The higher strength indicates more correlation between activities, so we define Strength(A_i, A_j, k) in terms of the following recursive formula in **Fig. 5**. **Fig. 6** shows the pseudo code of step 2 and **Fig. 7** an example of step 2 where **Fig. 7(a)** shows a strength DSM and **Fig. 7(b)** corresponding results of **Fig. 7(a)** after obtaining the strength for each pair of activities. After doing so, major activities

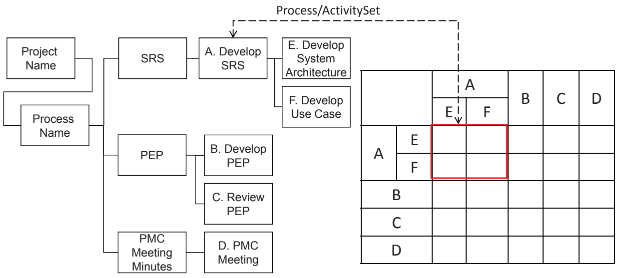


Fig. 8. Transforming WBS to DSM.

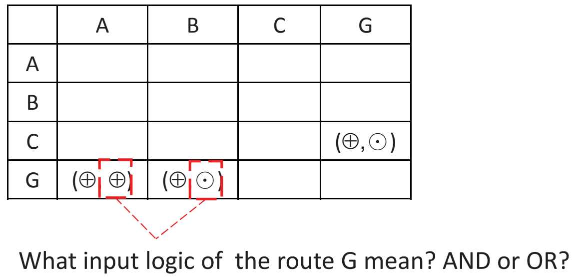


Fig. 9. Logic verification in DSM.

outputting work products can be identified. We use the DSM to conduct clustering based on these major activities, which are initial clusters. The clustering algorithm then groups other activities into clusters based on their strength relationships.

2.1.4. Organizing the WBS

The Project Management Institute (PMI) recommends a deliverable-oriented WBS hierarchy for project planning and control [16]. The project name is placed on level 1 and level 2 for processes included in the project. Level 3 is deliverables delivered by the parent process level. We recommend placing work products and system components on level 3. On level 4, activities or tasks are clustered for the deliverable level. The path searching partition algorithm [17] is applied to rearrange elements for each WBS level in this order.

2.2. WBS2Process

Project managers may edit the WBS for project planning, cost estimation, resources assignment, etc., but constraints exist in editing the WBS because project managers should follow project-defined processes to lead project execution. WBS editing constraints are suggested as follows:

- Project managers can add a project-specific work product or task.
- Project managers must assign resources to a task
- Tasks must be scheduled by the project manager
- Project managers cannot delete a deliverable or a task existing in the project-defined process. Deletion is only conducted if it is allowed in tailoring guidelines.
- Project managers cannot rename a work product or a task.

2.2.1. Transforming the WBS into a DSM

Although element types are defined for each WBS level, ambiguity remains while the WBS is being transformed into a DSM for elements that project managers add on WBS level 4 or break down into level 5, where elements types for these newly added elements may be overlooked. Project managers must identify WBS elements types in extension attributes when adding new elements to a WBS, and only activity element types are transformed into a DSM. **Fig. 8** shows a WBS and its corresponding DSM. The WBS contains six activities – A. Develop SRS and its child activities, E. Develop System Architecture, and F. Develop the Use Case, B. Develop PEP, C. Review PEP, and D. PMC Meeting.

Note that changes in element types impacts on DSM representation. After project managers break down activity A into activities E and activity F, for example, the element type of activity A should be changed to SubFlow or BlockActivity. The original DSM contains four activities and the revised DSM five and one BlockActivity A. Activities E and F are grouped in an ActivitySet named block1. The ActivitySet block1 is invoked by BlockActivity A.

2.2.2. Verifying and Merging the New and Original DSMs

Figure 9 shows a DSM with input and output logic. An activity has one input logic and one output logic. In **Fig. 9**, the input logic of G comes from outgoing A and outgoing B. The type of input logic of G, however, differs from A and B in the DSM. One is AND-Join and the other is OR-Join, so it is confusing to determine what the input logic of G is.

The same problem arises in output logic in a DSM. Input and output logic are verified by checking the same symbol logic for each column and row in a DSM. Merge the original DSM and new DSM starting in ActivityId mapping. ActivityId in the new DSM can be found in the original DSM only if the activity with the ActivityId is transformed from the project-defined process. Input and output logic of mapped activities in the new DSM are verified based on the original DSM. The verified result is placed in the new DSM, so the new DSM is the merged result and is ready to generate an XPDL format.

2.2.3. Generating Executable Process

The executable process derives from the merged DSM and XPDL file of the project-defined process. To exe-

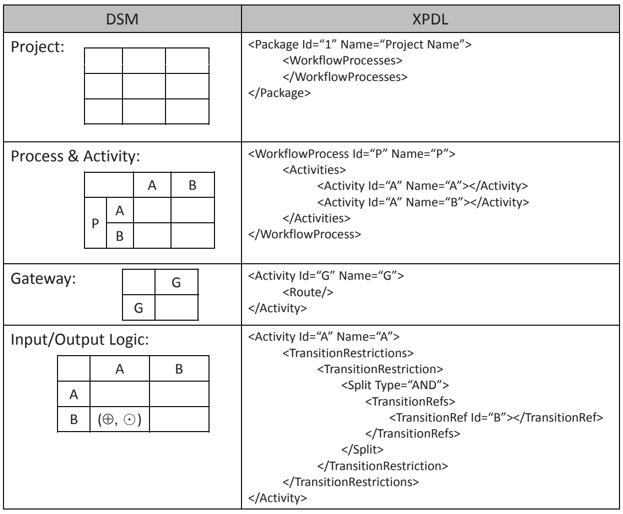


Fig. 10. Mapping from DSM to XPDL.

cute the process in a workflow engine, data and application definition in the XPDL of the project-defined process are needed to generate an executable process in XPDL format. **Fig. 10** shows mapping from the DSM to XPDL.

A DSM is used to represent a project, so corresponding XPDL tag <Package> is created in an XPDL file. There may be multiple processes in a project, such as requirements management, measurement and analysis, and project monitor and control processes. A process should involve activities for achieving the business goal and deliver work products for project monitoring and control. A process is mapped to tag <WorkflowProcess>, and an activity is mapped to tag <Activity>. <Activity> is a subelement of <WorkflowProcess>. A Gateway is an <Activity> having subelement <Route>.

Input and output logic are mapped to tag <TransitionRestriction> that is a subelement of an activity. The two tags, <Split> and <Join>, are the subelement of <TransitionRestriction>. In the subelement of tag <Activity>, <Split> indicates the output logic of the activity and <Join> its input logic.

Information contained in a DSM is not enough to execute a process. Project-specific information should be used to generate an executable XPDL for process execution. A WBS includes three project-specific types of information, which should be saved as a subelement of tag <Activity> in an XPDL file. The resource assigned in a WBS is mapped to tag <Performer> and the estimated task duration is saved in tag <Duration>, i.e., a subelement of <TimeEstimation>. Deliverables in a WBS should be recorded in tag <Artifact> and referenced by an activity in tag <Output> with attribute "ArtifactId."

3. Exemplary Scenario

In presenting a sample Project Management Process (PMP), for clarity, we simplify the example to explain how our approach can be realized systematically.

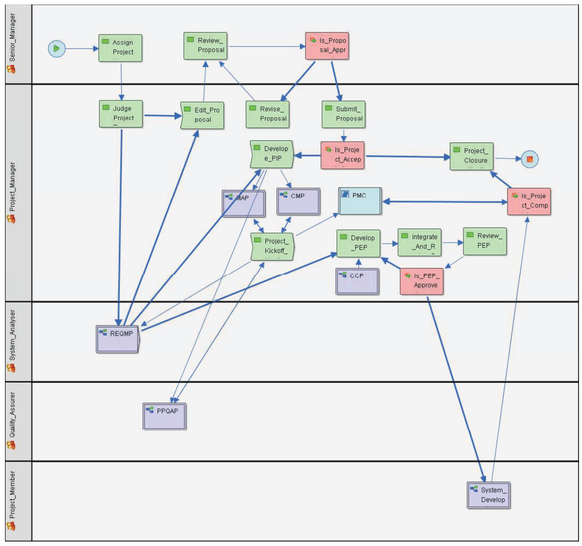


Fig. 11. Project management process.

3.1. Process2WBS Scenario

1. Representing the process using the DSM: The purpose of the PMP, as shown in **Fig. 11**, is to manage and control project execution, which includes five roles – senior manager, project manager, system analyst, quality guarantor, and project member. The PMP starts by assigning a project manager from a senior manager, than the project manager judges the project type for different execution flows. In the PMP, subprocesses such as REQMP, PPQAP, MAP, CMP, and CCP are represented as activities. The DSM represents subprocesses and their activities. An <ActivitySet>, such as PMC, and its activities are modeled as activities in a DSM. The XPDL file captures activities' deliverables and input/output not shown in **Fig. 11**.

2. Evaluating strength between activities: The DSM includes 26 activities. Here we model a subprocess as an activity with ActivityType="SubFlow" and evaluate degrees of strength based on the scores defined by our definition. **Fig. 12** shows the corresponding DSM of the PMP after strength assignment. After completing strength evaluation, major deliverable activities are identified to follow the cluster algorithm.

3. Clustering activities based on work products: After evaluating strength between activities in the PMP, the strength DSM is calculated by a macro-function in MS Excel based on the strength derivation algorithm proposed in **Fig. 6**. **Fig. 13** shows the strength DSM after running the strength derivation algorithm. If seven deliverables and corresponding major activities – activity Nos. 6, 8, 24, 20, 15, 16, and 10 – are identified by a project manager, then activities in the PMP are grouped based on the strength DSM in **Fig. 13**. Initial clusters are created for major activities. A cluster contains only one major activity, so

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Assign_Project_Manager	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Review_Proposal	2	0	0	0	3	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Is_Proposal_Approved?	3	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Revise Proposal	4	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Judge_Project_Type	5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Edit_Proposal	6	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0
Submit_Proposal	7	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Develop_PIP	8	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0
Is_Project_Accepted?	9	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Project_Closure_Meeting	10	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
MAP	11	0	0	0	0	0	0	0	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
CMP	12	0	0	0	0	0	0	0	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Project Kickoff Meeting	13	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Judge_Review_Type?	14	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0
Progress_Review_Meeting	15	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Milestone Review Meeting	16	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Manage_Action_Item	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3	0	0	0	0	0	0	0	0	0	0
Is_Project_Complete?	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
CCP	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Develop_PEP	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	2	0	0
Integrate_And_Revise_PEP	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0
Review_PEP	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0
Is_PEP_Approved?	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0
REQMP	24	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
System Develop Process	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
PPQAP	26	0	0	0	0	0	0	0	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 12. Project management process DSM.

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Assign_Project_Manager	1	1	0.111	0.111	0.111	0.333	0.111	0.111	0.074	0.037	0.012	0.049	0.049	0.037	0.012	0.004	0.004	0.004	0.008	0.025	0.0741	0.0741	0.0741	0.074	0.111	0.025	0.0494
Review_Proposal	2	0.11	1	1	1	0.333	1	1	0.444	0.333	0.111	0.296	0.296	0.222	0.074	0.025	0.025	0.025	0.049	0.148	0.4444	0.4444	0.4444	0.444	0.667	0.148	0.2963
Is_Proposal_Approved?	3	0.11	1	1	1	0.333	1	1	0.444	0.333	0.111	0.296	0.296	0.222	0.074	0.025	0.025	0.025	0.049	0.148	0.4444	0.4444	0.4444	0.444	0.667	0.148	0.2963
Revise Proposal	4	0.11	1	1	1	0.333	1	1	0.444	0.333	0.111	0.296	0.296	0.222	0.074	0.025	0.025	0.025	0.049	0.148	0.4444	0.4444	0.4444	0.444	0.667	0.148	0.2963
Judge_Project_Type	5	0.33	0.333	0.333	0.333	1	0.333	0.333	0.222	0.111	0.037	0.148	0.148	0.111	0.037	0.012	0.012	0.012	0.025	0.074	0.2222	0.2222	0.2222	0.222	0.333	0.074	0.1481
Edit_Proposal	6	0.11	1	1	1	0.333	1	1	0.444	0.333	0.111	0.296	0.296	0.222	0.074	0.025	0.025	0.025	0.049	0.148	0.4444	0.4444	0.4444	0.444	0.667	0.148	0.2963
Submit_Proposal	7	0.11	1	1	1	0.333	1	1	0.444	0.333	0.111	0.296	0.296	0.222	0.074	0.025	0.025	0.025	0.049	0.148	0.4444	0.4444	0.4444	0.444	0.667	0.148	0.2963
Develop_PIP	8	0.07	0.444	0.444	0.444	0.222	0.444	0.444	1	0.333	0.111	0.667	0.667	0.222	0.074	0.025	0.025	0.025	0.049	0.148	0.4444	0.4444	0.4444	0.444	0.667	0.148	0.6667
Is_Project_Accepted?	9	0.04	0.333	0.333	0.333	0.111	0.333	0.333	0.333	1	0.333	0.222	0.222	0.074	0.037	0.037	0.037	0.037	0.111	0.049	0.1481	0.1481	0.1481	0.148	0.222	0.049	0.2222
Project_Closure_Meeting	10	0.01	0.111	0.111	0.111	0.037	0.111	0.111	0.111	0.333	1	0.074	0.074	0.037	0.111	0.111	0.111	0.111	0.333	0.016	0.0494	0.0494	0.0494	0.049	0.074	0.111	0.0741
MAP	11	0.05	0.296	0.296	0.296	0.148	0.296	0.296	0.667	0.222	0.074	1	0.444	0.333	0.111	0.037	0.037	0.037	0.037	0.099	0.2963	0.2963	0.2963	0.296	0.444	0.099	0.4444
CMP	12	0.05	0.296	0.296	0.296	0.148	0.296	0.296	0.667	0.222	0.074	0.444	1	0.333	0.111	0.037	0.037	0.037	0.037	0.099	0.2963	0.2963	0.2963	0.296	0.444	0.099	0.4444
Project Kickoff Meeting	13	0.04	0.222	0.222	0.222	0.111	0.222	0.222	0.222	0.074	0.037	0.333	0.333	1	0.333	0.111	0.111	0.111	0.111	0.074	0.2222	0.2222	0.2222	0.222	0.333	0.074	0.3333
Judge_Review_Type?	14	0.01	0.074	0.074	0.074	0.037	0.074	0.074	0.074	0.037	0.111	0.111	0.111	0.333	1	0.333	0.333	0.333	0.333	0.025	0.0741	0.0741	0.0741	0.074	0.111	0.111	0.1111
Progress_Review_Meeting	15	0	0.025	0.025	0.025	0.012	0.025	0.025	0.025	0.037	0.111	0.037	0.037	0.111	0.333	1	1	1	0.333	0.012	0.037	0.037	0.037	0.037	0.037	0.111	0.037
Milestone Review Meeting	16	0	0.025	0.025	0.025	0.012	0.025	0.025	0.025	0.037	0.111	0.037	0.037	0.111	0.333	1	1	1	0.333	0.012	0.037	0.037	0.037	0.037	0.037	0.111	0.037
Manage_Action_Item	17	0	0.025	0.025	0.025	0.012	0.025	0.025	0.025	0.037	0.111	0.037	0.037	0.111	0.333	1	1	1	0.333	0.012	0.037	0.037	0.037	0.037	0.037	0.111	0.037
Is_Project_Complete?	18	0.01	0.049	0.049	0.049	0.025	0.049	0.049	0.049	0.111	0.333	0.037	0.037	0.111	0.333	0.333	0.333	0.333	1	0.037	0.1111	0.1111	0.1111	0.111	0.074	0.333	0.037
CCP	19	0.02	0.148	0.148	0.148	0.074	0.148	0.148	0.148	0.049	0.016	0.099	0.099	0.074	0.025	0.012	0.012	0.012	0.037	1	0.3333	0.3333	0.3333	0.333	0.222	0.111	0.0988
Develop_PEP	20	0.07	0.444	0.444	0.444	0.222	0.444	0.444	0.444	0.148	0.049	0.296	0.296	0.222	0.074	0.037	0.037	0.037	0.111	0.333	1	1	1	1	0.667	0.333	0.2963
Integrate_And_Revise_PEP	21	0.07	0.444	0.444	0.444	0.222	0.444	0.444	0.444	0.148	0.049	0.296	0.296	0.222	0.074	0.037	0.037	0.037	0.111	0.333	1	1	1	1	0.667	0.333	0.2963
Review_PEP	22	0.07	0.444	0.444	0.444	0.222	0.444	0.444	0.444	0.148	0.049	0.296	0.296	0.222	0.074	0.037	0.037	0.037	0.111	0.333	1	1	1	1	0.667	0.333	0.2963
Is_PEP_Approved?	23	0.07	0.444	0.444	0.444	0.222	0.444	0.444	0.444	0.148	0.049	0.296	0.296	0.222	0.074	0.037	0.037	0.037	0.111	0.333	1	1	1	1	0.667	0.333	0.2963
REQMP	24	0.11	0.667	0.667	0.667	0.333	0.667	0.667	0.667	0.222	0.074	0.444	0.444	0.333	0.111	0.037	0.037	0.037	0.074	0.222	0.6667	0.6667	0.6667	0.667	1	0.222	0.4444
System Develop Process	25	0.02	0.148	0.148	0.148	0.074	0.148	0.148	0.148	0.049	0.111	0.099	0.099	0.074	0.111	0.111	0.111	0.111	0.333	0.111	0.3333	0.3333	0.3333	0.333	0.222	1	0.0988
PPQAP	26	0.05	0.296	0.296	0.296	0.148	0.296	0.296	0.667	0.222	0.074	0.444	0.444	0.333	0.111	0.037	0.037	0.037	0.037	0.099	0.2963	0.2963	0.2963	0.296	0.444	0.099	1

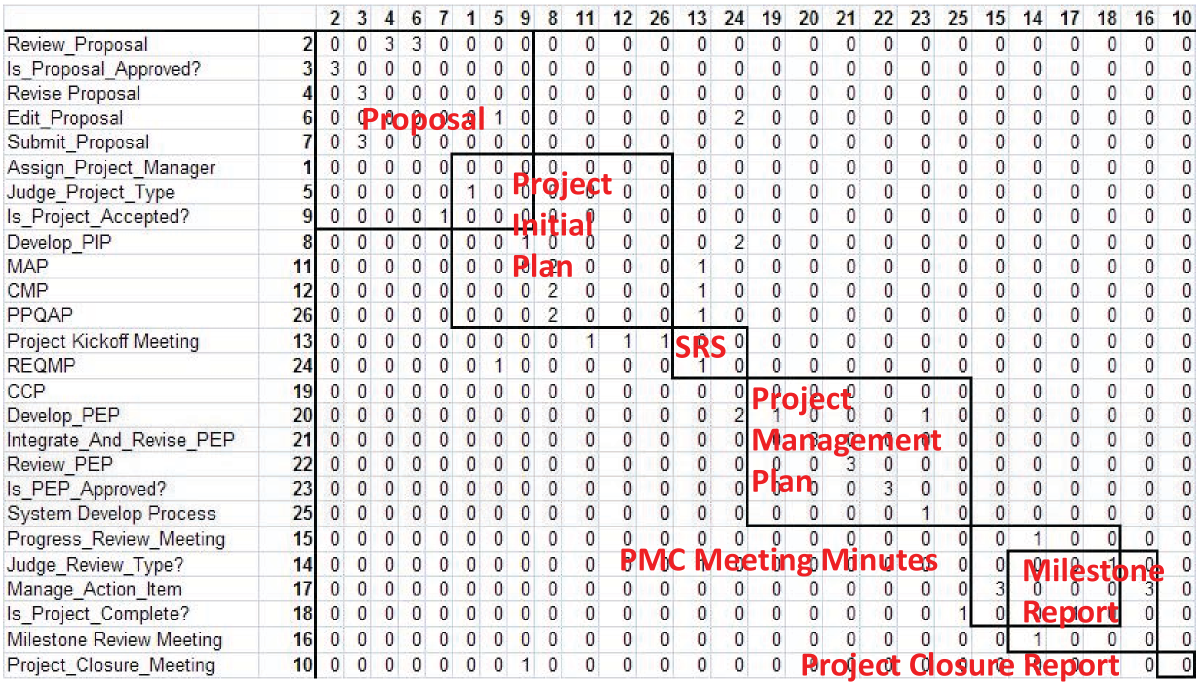
Fig. 13. DSM after strength derivation.

cluster Nos. 1-7 are created for activity Nos 6, 8, 24, 20, 15, 16, and 10. Other activities, not major activities, must join clusters based on strength values between activities and major activities. The higher the strength value indicates higher dependence between an activity and cluster. Strength values between an activity and each major activity are compared and the cluster with the highest strength is selected to group an activity, e.g., in **Fig. 13**, activity Assign_Project_Manager joins clusters 1 and 3 with the highest strength 0.111 and activity Review_Proposal joins cluster 1 with highest strength value 1. **Fig. 14** shows the DSM result after clustering. Seven deliverables are identified in clustered DSM. Activity Nos. 2, 3, 4, 6, 7, 1, 5, and 9 contribute to the proposal. Activity Nos. 1, 5, 9, 8, 11, 12, and 26 contribute to the project initial plan. The clustering result indicates that the activity Nos. 1, 5, and 9 are shared in the lifecycle of the proposal and project initial plan. The clustered DSM represents clusters based on strength analysis, but the DSM cannot represent clusters and sequence of activities for each cluster simultaneously if there are activities shared between two clusters.

4. Organizing the WBS: The WBS in **Fig. 15** is organized based the clustered result. The project manager identifies activity Nos. 6, 8, 24, 20, 15, 16, and 10 as major activities. Activity 6 delivers the proposal and activity 8 delivers the project initial plan. The SRS is delivered by activity 24 and activity 20 produces the project management plan. Activity 15 outputs project meeting minutes and milestone report is delivered by activity 16. The WBS is organized from PMP alone and the subprocess is not represented in **Fig. 15**. Subprocesses such as REQMP, MAP, and CMP are a posited sibling of the project management process on level 2, but the process on WBS level 2 still must be rearranged based on DSM partitioning analysis.

3.2. WBS2Process Scenario

For better process control and to improve process execution efficiency, a WBS with project-specific information must be transformed into an executable process that



		2	3	4	6	7	1	5	9	8	11	12	26	13	24	19	20	21	22	23	25	15	14	17	18	16	10
Review_Proposal	2	0	0	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Is_Proposal_Approved?	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Revise Proposal	4	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Edit_Proposal	6	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
Submit_Proposal	7	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Assign_Project_Manager	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Judge_Project_Type	5	0	0	0	0	0	1	0	[illegible]	[illegible]	[illegible]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Is_Project_Accepted?	9	0	0	0	0	1	0	0	[illegible]	[illegible]	[illegible]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Develop_PIP	8	0	0	0	0	0	0	0	1	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
MAP	11	0	0	0	0	0	0	0	[illegible]	[illegible]	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
CMP	12	0	0	0	0	0	0	0	0	2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
PPQAP	26	0	0	0	0	0	0	0	0	2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Project Kickoff Meeting	13	0	0	0	0	0	0	0	0	0	1	1	1	[illegible]	0	0	0	0	0	0	0	0	0	0	0	0	0
REQMP	24	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
CCP	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	[illegible]	[illegible]	[illegible]	0	0	0	0	0	0	0	0	0
Develop_PEP	20	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	[illegible]	[illegible]	0	1	0	0	0	0	0	0	0
Integrate_And_Revise_PEP	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0	0	0	0
Review_PEP	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	[illegible]	0	3	0	0	0	0	0	0	0	0	0
Is_PEP_Approved?	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	[illegible]	[illegible]	0	3	0	0	0	0	0	0	0	0
System Develop Process	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
Progress_Review_Meeting	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Judge_Review_Type?	14	0	0	0	0	0	0	0	0	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	[illegible]	[illegible]	[illegible]	[illegible]	0
Manage_Action_Item	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	3	0
Is_Project_Complete?	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	[illegible]	[illegible]	[illegible]	0	0
Milestone Review Meeting	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Project_Closure_Meeting	10	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0

Fig. 14. Clustered DSM of project management process.

		任務名稱
1		**Project Name**
2		**Process Management Process**
3		**Proposal**
4		Assign_Project_Manager
5		Judge_Project_Type
6		Edit_Proposal
7		**Block1**
8		Review_Proposal
9		Is_Proposal_Approved?
10		Revise Proposal
11		Submit_Proposal
12		Is_Project_Accepted?
13		**Project Initial Plan**
14		Develop_PIP
15		**SRS**
16		Project Kickoff Meeting
17		**Project Management Plan**
18		Develop_PEP
19		Integrate_And_Revise_PEP
20		Review_PEP
21		Is_PEP_Approved?
22		**PMC Meeting Minutes**
23		Judge_Review_Type?
24		Progress_Review_Meeting
25		Manage_Action_Item
26		Is_Project_Complete?
27		**Milestone Report**
28		Milestone Review Meeting
29		**Project Closure Report**
30		Project_Closure_Meeting

Fig. 15. WBS of project management process.

```
<Activity Id="Develop_PEP" Name="Develop_PEP">
    <Implementation>
        <No/>
    </Implementation>
    <Performer>Project_Manager:Steven</Performer>
    <StartMode>
        <Manual/>
    </StartMode>
    <FinishMode>
        <Manual/>
    </FinishMode>
    <SimulationInformation>
        <Cost>64</Cost>
        <TimeEstimation>
            <WaitingTime>2</WaitingTime>
            <WorkingTime>30</WorkingTime>
            <Duration>32</Duration>
        </TimeEstimation>
    </SimulationInformation>
    <TransitionRestrictions>
        <TransitionRestriction>
            <Join Type="XOR"/>
        </TransitionRestriction>
    </TransitionRestrictions>
</Activity>
<Activity Id="PPQAP">
    <Implementation>
        <SubFlow Execution="ASYNCHR" Id="PPQA_Process"/>
    </Implementation>
    <TransitionRestrictions>
        <TransitionRestriction>
            <Join Type="XOR"/>
        </TransitionRestriction>
    </TransitionRestrictions>
</Activity>
```

Fig. 16. Partial XPDL of PMP.

can be executed automatically by the workflow engine. To ensure consistency between the WBS and project-defined process, the project manager should apply WBS editing constraints in Section 2.2 to assign resources and plan a schedule in the WBS template generated in Process2WBS. After finishing WBS editing, the project manager starts transforming the WBS into a DSM using the domain mapping table in **Fig. 2**. Relationships and input/output logic of the DSM produced by WBS2Process are verified based on the original DSM produced by Process2WBS. These two DSMs are then merged into a new DSM to be used to generate an executable process with project-specific information in XPDL format based on the mapping table in **Fig. 10**. **Fig. 16** shows partial XPDL of the project management process derived from WBS2Process. Activity Develop_PEP contains project-specific information, i.e., performer and simulation information, to be used by the workflow engine to execute the project management process automatically.

4. Related Work

The DSM has been applied to several categories, including building construction [18–23], semiconductors [24, 25], automotive [26–28], aerospace [29–33], telecom [34], and electronics [35] industries. This section lists related work for process and project integration.

Christoph Bussler [36] stated that the main reason for PM tool failures, e.g., out of date or incorrect schedule, is synchronization missing between the project plan and actual execution tasks. He integrated WfMS with the project management tool in two parts – schema integration to map conceptual objects of WfMS and PM onto each other and behavior integration to define the scenario and interfaces among the user, WfMS, and PM when changing data. This study does not address dependence between WfMS and organization's set of standard processes because the project process should follow the organization's set of standard processes and constraints by criteria and tailoring guidelines.

Michael Gnatz et al. [37] concluded that most development projects have complex dependence among tasks, and less experienced project managers often under-estimate schedules and efforts. They stated that well-defined and repeatable processes offer knowledge and lessons to less experienced project managers and propose a process meta-model to constrain the instantiation of the process model for deriving the project plan.

Lawrance M.L. Chung and Keith C.C. Chan [38] addressed the limits of the Process Management Environments (PME) and Project Management Tools (PMT), e.g., PMEs do not provide a project schedule. They presented an integrated process and project management tool via the map process and project concept and provide an object function to minimize the project schedule.

Thibault Alexandre et al. [39] discuss process integration requirements based on product and manufacturing data. To reduce product and process design time and cost, they provide a process plan schema with degrees of freedom and rules on transformation to integrate the project process based on product data.

Ali Bahrami [40] proposed integrated process management integrating project management, business process modeling, simulation, and workflow to support scheduled workflow execution. The purpose is to generate a workflow based on a scheduling tool. The system exports the workflow process in XPDL format. The following defines three activity types:

- Simple Task: an activity including one task
- Hierarchical Task: an activity including several tasks that did not previously exist
- Process Component: an activity including several tasks that previously existing tasks

Table 1. Comparison of research on process and project integration.

	Christoph Bussler [36]	Michael Gnatz et al. [37]	Lawrance M.L. Chung and Keith C.C. Chan [38]	Thibault Alexandre et al. [39]	Ali Bahrami [40]	Our Approach
Domain Concept Mapping	Yes	Yes	Yes	No	No	Yes
Transformation between Process and Project	Bi-direction	One way	One way	One way	Bi-direction	Bi-direction
Activity Input/Output Logic Support	No	No	Yes	Yes	No	Yes
Feedback and Cyclic Support	Not mentioned	Not mentioned	Yes	Yes	Not mentioned	Yes

However, no domain concepts are mapped between project and process.

We compare these process and project integration approaches with a list of criteria in **Table 1**, detailed below.

- Domain concept mapping: Process concepts and project concepts differ and need mapping to clarify concepts. Is there any mapping, such as domain mapping table, between process concepts and project concepts?
- Transformation between process and project: Changes in a process (project) should be synchronized with the project (process) to improve consistency and maintainability. Is transformation between process and project bi-directional, from process to project, and from project to process, or one-way?
- Activity input/output logic support: Input/output logic controls the flow of activities and affects both processes, i.e., activity dependence, and project, i.e., project schedule. Does the integration approach support input/output logic during transformation?
- Feedback and cyclic support: Feedback and cyclic are common relationships in processes. Missing feedback and cycles adversely affect the accuracy of project schedules. Does the approach support feedback and cycles?

5. Conclusions and Projected Work

We have proposed a DSM-based approach for integrating a process with the WBS. The WBS template is derived from a project-defined process and increases WBS conformity with the project-defined process. The domain-mapping table mapped between a process and the DSM, and the DSM and WBS helps correct mapping concepts between a process and the WBS. Our clustering algorithm analyzes strength among activities to group activities based on deliverables. WBS2Process generates the executable process in XPDL format.

Our projected work is three-focus:

- Enabling tailoring from the organization's set of standard processes to project-defined processes based on criteria and tailoring guidelines.
- Enhancing consistency verification between the project-defined process and the executable process by applying process compliance measurement and analysis.
- Evaluating and improving process performance, measureable concepts such as process compliance, process efficiency, and process effectiveness, corresponding measures, and corresponding metrics called process performance metrics are needed to develop and collect during project execution. Process performance is then evaluated based on process performance metrics.

References:

[1] "Work Breakdown Structure "WBS"," http://www.hyperthot.com/pm_wbs.htm/

[2] Booz, Allen, and Hamilton, "Earned Value Management Tutorial Module 2, Work Breakdown Structure," Office of Project Assessment, 2008.

[3] NASA 2001, "NASA NPR 9501.2D.," May 23, 2001.

[4] R. A. Radice, N. K. Roth, A. C. O'Hara, Jr., and W. A. Ciarfella, "A Programming Process Architecture," j-IBM-SYS-J, Vol.24, No.2, pp. 79-90, 1985.

[5] D. V. Steward, "The Design Structure System, A Method for Managing the Design of Complex Systems," IEEE Trans. on Engineering Management, Vol.28, pp. 71-74, 1981.

[6] T. R. Browning, "Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions," IEEE Trans. on Engineering Management, Vol.48, No.3, pp. 292-306, 2001.

[7] T. R. Browning and S. D. Eppinger, "Modeling Impacts of Process Architecture on Cost and Schedule Risk in Product Development," IEEE Trans. on Engineering Management, Vol.49, No.4, pp. 428-442, 2002.

[8] A. A. Yassine, D. E. Whiteny, and T. Zambito, "Assessment of Rework Probabilities for Simulating Product Development Processes Using the Design Structure Matrix," Proc. of the DETC 01: ASME 2001 Int. Design Engineering Technical Conf., Pittsburgh PA, 2001.

[9] "XML Process Definition Language "XPDL"," http://www.wfmc.org/xpdl.html/

[10] J. Lee and Y.-Y. Fanjiang, "Modeling Imprecise Requirements with XML," Information and Software Technology, Vol.45, No.7, pp. 445-460, 2003.

[11] J. Lee and K.-H. Hsu, "Modeling Software Architectures with Goals in Virtual University Environment," Information and Software Technology, Vol.44, No.6, pp. 361-380, 2002.

[12] J. Lee, C.-L. Wu, W.-T. Lee, and K.-H. Hsu, "Aspect-Enhanced Goal-Driven Sequence Diagram," Int. J. of Intelligent Systems, Vol.25, pp. 712-732, 2010.

[13] J. Lee and N.-L. Xue, "Analyzing User Requirements by Use Cases, A Goal-Driven Approach," IEEE Software, Vol.16, No.4, pp. 92-101, 1999.

[14] J. Lee, N.-L. Xue and J.-Y. Kuo, "Structuring Requirements Specifications with Goals," Information and Software Technology, Vol.43, No.2, pp. 121-135, 2001.

[15] W.-T. Lee, W.-Y. Deng, J. Lee, and S.-J. Lee, "Change Impact Analysis with a Goal-Driven Traceability-Based Approach," Int. J. of Intelligent Systems, Vol.25, pp. 878-908, 2010.

[16] G. T. Haugan, "Effective Work Breakdown Structures," Management Concepts, 2001.

[17] D. V. Steward, "Partitioning and Tearing Systems of Equations," J. of the Society for Industrial and Applied Mathematics, Series B, Numerical Analysis, Vol.2, No.2, pp. 345-365, 1965.

[18] "Application of the Analytical Design Planning Technique to Construction Project Management," Project Manage. J., Vol.31, pp. 48-59, 2000.

[19] S. Austin, A. Baldwin, B. Li, and P. Waskett, "Development of the ADePT Methodology: An Interim Report on the Link IDAC 100 Project," Loughborough University, Dept. of Civil and Building Engineering, Loughborough, U. K., 1998.

[20] S. Austin, A. Baldwin, and A. Newton, "A Data Flow Model to Plan and Manage the Building Design Process," J. Eng. Des., Vol.7, pp. 3-25, 1996.

[21] P. Huovila, L. Koskela, L. Pietilainen, and V.-P. Tanhuanpaa, "Use of the Design Structure Matrix in Construction," In 3rd Int. Workshop on Lean Construction, Albuquerque, NM, 1995.

[22] K. Kahkonen, V.-P. Tanhuanpaa, and S. Leino, "Design Process Analysis, Optimization and Management, XA Practical Tool for the Construction and Engineering Projects," VTT Building Technology, Finland, 1997.

[23] L. Koskela, G. Ballard, and V.-P. Tanhuanpaa, "Toward Lean Design Management," In 5th Annu. Conf. of the International Group for Lean Construction "IGLC-5," 1997.

[24] "Innovation at the speed of information," In Harvard Bus. Rev., Vol.79, pp. 149-V158, 2001.

[25] S. M. Osborne, "Product Development Cycle Time Characterization Through Modeling of Process Iteration," M.S. thesis, MIT, Cambridge, MA, 1993.

[26] J. Malmstrom, P. Pikosz, and J. Malmqvist, "The Complementary Roles of IDEF0 and DSM for the Modeling of Information Management Processes," Concurrent Eng.: Res. Applicat., Vol.7, pp. 95-V103, 1999.

[27] G. J. Rushton and A. Zakarian, "Modular Vehicle Architectures, A Systems Approach," In 10th Annu. Int. Symp. of INCOSE, Minneapolis, MN, 2000, pp. 29-35, 2000.

[28] M. W. Sequeira, "Use of the Design Structure Matrix in the Improvement of an Automobile Development Process," M.S. thesis, MIT, Cambridge, MA, 1991.

[29] P. J. Clarkson and J. R. Hamilton, "'Signposting': A Parameter-Driven Task-Based Model of the Design Process," Res. Eng. Des., Vol. 12, pp. 18-38, 2000.

[30] M. L. Danilovic, "Leadership and Organization of Integration in Product Development," Ph.D. dissertation, Linkopings Universitet, Linkoping, Sweden, 1999.

[31] B. J. Makins and D. W. Miller, "Web-Based Aerospace System Evaluation Software: The Development and Assessment of Conceptual Space Missions," In Proc. 10th Annu. Int. Symp. of INCOSE, Minneapolis, MN, 2000, pp. 167-174, 2000.

[32] M. Nour and J. Scanlan, "Modeling and Simulating Product Development Process," In Proc. 6th Int. Conf. on Concurrent Enterprising, Toulouse, France, 2000, pp. 111-118, 2000.

[33] R. H. Ahmadi, T. A. Roemer, and R. H. Wang, "Structuring Product Development Processes," Eur. J. Oper. Res., Vol.130, pp. 539-558, 2001.

[34] R. D. Pinkett, "Product Development Process Modeling and Analysis Digital Wireless Telephones," S. M. thesis, MIT, Cambridge, MA, 1998.

[35] M. Carrascosa, S. D. Eppinger, and D. E. Whitney, "Using the Design Structure Matrix to Estimate Product Development Time," In Proc. ASME Des. Eng. Tech. Conf., "Design Automation Conf.," Atlanta, GA, 1998.

[36] C. Bussler, "Workflow Instance Scheduling with Project Management Tools," 9th Int. Workshop on Database and Expert Systems Applications "DEXA'98," p. 753, 1998.

[37] M. Gnatz, M. Deubler, M. Meisinger, and A. Rausch, "Towards an Integration of Process Modeling and Project Planning," 5th Int. Workshop on Software Process Simulation and Modeling "ProSim 2004," pp. 22-31, 2004.

[38] L. M. L. Chung and K. C. C. Chan, "Integrating Project Planning and Process Modeling for Software Development," IEEE Workshop on Application-Specific Software Engineering and Technology, IEEE Computer Society, Los Alamitos, CA, USA, 1999, p. 276, 1999,

[39] A. Thibault, A. Siadat, R. Bigot, and P. Martin, "Proposal for Product Process Integration using Classification and Rules," In EUROCON, 2007. The Int. Conf. on "Computer as a Tool," 9-12, 2007, pp. 753-758, 2007,

[40] A. Bahrami, "Integrated Process Management, From Planning to Work Execution," In BSN '05: Proc. of the IEEE EEE05 int. workshop on Business services networks, Hong Kong, IEEE Press, Piscataway, NJ, USA, p. 11, 2005.

Name:
Jonathan Lee

Affiliation:
Department of Computer Science and Information Engineering, National Central University

Address:
No.300, Jhongda Rd., Jhongli City, Taoyuan County 320, Taiwan

Brief Biographical History:
1993 Ph.D. degree from Texas A&M University
1993- The faculty of the Department of Computer Science and Information Engineering at National Central University (NCU) in Taiwan

Main Works:
• agent-based software engineering, service-oriented computing, goal-driven software engineering

Name:
Shin-Jie Lee

Affiliation:
Department of Computer Science and Information Engineering, National Central University

Address:
No.300, Jhongda Rd., Jhongli City, Taoyuan County 320, Taiwan

Brief Biographical History:
2007 Ph.D. in computer science and information engineering from National Central University (NCU) in Taiwan
2007- Postdoctoral Researcher in Software Research Center at NCU

Main Works:
• agent-based software engineering, service-oriented computing, object-oriented software engineering

Name:
Whan-Yo Deng

Affiliation:
Department of Computer Science and Information Engineering, National Central University

Address:
No.300, Jhongda Rd., Jhongli City, Taoyuan County 320, Taiwan

Brief Biographical History:
2010- Ph.D. student in the department of Computer Science and Information Engineering at National Central University (NCU) in Taiwan

Main Works:
• project planning, project management, process management

Name:
Kuo-Hsun Hsu

Affiliation:
Assistant Professor of Department of Computer and Information Science, National Taichung University

Address:
140 Min-Sheng Rd., Taichung City, Taiwan 403

Brief Biographical History:
1992-1996 B.S. degree from Computer and Information Science, National Chiao Tung University, Taiwan.
1997-2003 Ph.D. degree from Computer Science and Information Engineering, National Central University, Taiwan

Main Works:
• software engineering, requirement engineering, software architecture, service-orient architecture, CMMI

Name:
Wen-Tin Lee

Affiliation:
Department of Computer Science and Information Engineering, National Central University

Address:
No.300, Jhongda Rd., Jhongli City, Taoyuan County 320, Taiwan

Brief Biographical History:
2007 Ph.D. in computer science and information engineering from National Central University (NCU) in Taiwan
2007- Postdoctoral Researcher in Software Research Center at NCU

Main Works:
• requirements engineering, software process improvement, service-oriented software engineering

Name:
Shang-Pin Ma

Affiliation:
Department of Computer Science and Engineering, National Taiwan Ocean University

Address:
2 Pei-Ning Road, Keelung ,Taiwan 20224, R.O.C.

Brief Biographical History:
1999 B.S. degrees in Computer Science and Information Engineering from National Central University, Chungli, Taiwan
2007 Ph.D. degrees in Computer Science and Information Engineering from National Central University, Chungli, Taiwan
2008- Assistant professor of Computer Science, and Engineering Department, National Taiwan Ocean University, Keelung, Taiwan

Main Works:
• software engineering, service-oriented computing, software process improvement