Composing Workflow Activities on the Basis of Data-Flow Structures

Han van der Aa[1], Hajo A. Reijers[1,2], and Irene Vanderfeesten[3]

[1] Dept. of Mathematics and Computer Science, Eindhoven University of Technology, PO Box 513, 5600MB Eindhoven, The Netherlands
[2] Perceptive Software, Piet Joubertstraat 4, 7315AV Apeldoorn, The Netherlands
[3] School of Industrial Engineering, Eindhoven University of Technology, PO Box 513, 5600MB Eindhoven, The Netherlands

Abstract. The proper composition of activities is important for the efficient execution of a workflow process. In this paper, an approach is presented that utilizes the data-flow underlying a workflow process to determine the importance and semantic relatedness of the various, elementary data-processing steps. Based on these aspects, fundamental guidelines are proposed to drive and objectify the task of activity composition in the context of workflow design.

1 Introduction

The bulk of business processes in the service domain pursue the production of an informational product, such as a mortgage contract, a decision on a damage claim, or a commercial offer. We will refer to these processes as *workflows*. When designing a workflow, one should carefully consider how to properly design its *activities* (or *tasks*). An activity is a *logical piece of work* within a workflow, which may comprise a number of elementary data processing steps. For example, the activity of calculating a mortgage amount may consist of entering the current interest rate, choosing the discount rate negotiated by the customer, and calculating the amortized amount of debt.

The focus of attention in this paper is on the grouping of elementary data processing steps into activities. We will refer to this act as *activity composition*. Proper composition can result in activities that have a proper size, i.e. are of a right *granularity*. They provide a balance between an increased number of work hand-overs that result from many small activities against the reduced flexibility caused by too many large activities [1]. Secondly, activity composition can be used to increase the meaningfulness of activities for employees executing these [2].

This paper introduces fundamental design guidelines for activity composition by exploiting the structural data-flow relations in a workflow. To make our ideas operational, we will assume that such relations are captured in a Product Data Model (PDM). However, the guidelines can be easily transferred to comparable data-flow specifications, e.g. the data flow matrices of [3]. The PDM that we

F. Daniel, J. Wang, and B. Weber (Eds.): BPM 2013, LNCS 8094, pp. 275–282, 2013.

build on stems from Product Based Workflow Design (PBWD), a method for the radical redesign of workflows [4]. It is proposed here that data-flow relations in the form of a PDM can be used to determine the *semantic relatedness* as well as the *relative importance* of its elements. These two notions are used to propose guidelines on activity composition, which form the main contribution of this paper.

This work extends the state of the art in several ways. The interaction between the steps in a process and the data being processed is at the basis of a wide range of research efforts, e.g. [3,5]. Yet, their exclusive focus is on the detection of data-flow errors. While we agree that it is important to ensure that a workflow works correctly, they leave open the issue of designing granular, meaningful tasks. The job design literature does address these issues [2,6], but we noted that the provided guidance is rather abstract and does not rest on a detailed understanding of the data-flow perspective in a specific process. Earlier, we proposed metrics to evaluate the quality of activity compositions on the basis of job design insights [7], but these can only be retrospectively applied on activities already composed. The guidelines we provide ensure a correct data processing through its reliance on a PDM, while they additionally lead to concrete and proper compositions of activities.

In the remainder of this paper, Sect. 2 provides an example to motivate the goal of activity composition and introduces some important notions. Sect. 3 proposes our solution: three guidelines that objectify activity composition. Finally, Sect. 4 concludes the paper with a discussion and directions for future research.

2 Motivating Example

To motivate the application of activity composition in a workflow process, this section presents the design of activities for an example case. The example, introduced in [8], considers the process that deals with requests for governmental student grants in the Netherlands. The presented process is a simplified version of the actual procedure as implemented by the "Dienst Uitvoering Onderwijs" (DUO)[1], the governmental agency in the Netherlands responsible for the assessment of student grant requests.

2.1 Product Data Model

Fig. 1 presents the PDM of the *student grants* example. A PDM contains a set of *data elements*, which are depicted as labelled circles. The top element $i42$, referred to as the *root* element, resembles the total student grant assigned to an applicant. The other data elements in the PDM are data elements that are relevant for the computation of $i42$, the ultimate goal of this workflow. A description of all data elements is provided in Table 1. The values for data elements for a specific case are computed by executing *operations* on data elements.

[1] See `http://www.duo.nl`

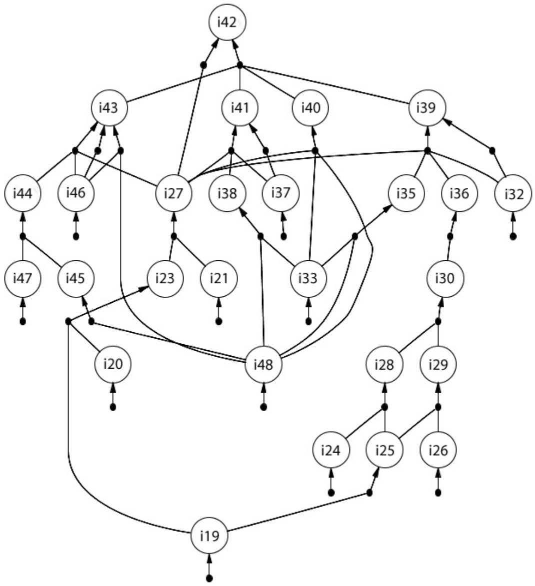

Fig. 1. Product Data Model of the student grants example

Operations are depicted as black dots in the figure. Each operation requires a set of input elements, and produces a single output data element. For example, Fig. 1 expresses that a value for $i42$ can be computed based on the values for data elements $i39$, $i40$, $i41$, and $i43$. These four elements represent the four types of student grant an applicant may be eligible to receive: (i) a basic grant ($i40$), (ii) a supplementary grant ($i39$), (iii) a loan ($i41$), and (iv) any credit for tuition fees ($i43$). A value for $i42$ can also be computed via a second operation, which has $i27$ as its only input element. This input element represents an applicant's eligibility to receive a grant[2]. The second operation is enabled when the value of $i27$ is negative, i.e. the applicant is not entitled to receive a grant. Hence, the execution of this operation means that an application is rejected. We refer to distinct operations that can produce a value for the same data element, as *alternative operations*; they provide alternative routings through a workflow. Finally, the figure shows operations that do not have input elements. These operations are referred to as *leaf operations*. Leaf operations produce values for *leaf elements*, i.e. data that is received from outside the process. In this example, the leaf elements contain the data that is provided by the applicant, such as $i20$, the applicant's date of birth. For a more detailed description of the case, the interested reader is referred to [8, p.193].

[2] An applicant must have the Dutch nationality (stored in $i21$) and may not be older than thirty ($i23$) in order to be eligible to receive a grant.

Table 1. Description of data elements present in the student grants example

ID	Description	ID	Description
i19	Date of request	i36	Parental contribution
i20	Birth date of applicant	i37	Requested amt. of loan
i21	Nationality of applicant	i38	Max. amt. of loan
i23	Age of applicant	i39	Amt. of supplementary grant assigned
i24	Social Security Number of father	i40	Amt. of basic grant assigned
i25	Reference year for tax authority	i41	Amt. of loan assigned
i26	Social Security Number of mother	i42	Total amt. of student grant assigned
i27	Applicant has the right to receive grant	i43	Amt. of tuition credit assigned
i28	Income of father of applicant	i44	Max. amt. of credit for tuition fees
i29	Income of mother of applicant	i45	Tuition fees of educational institution
i30	Income of parents of applicant	i46	Has requested credit for tuition fees
i32	Has requested a supplementary grant	i47	Tuition fees declared by law
i33	Living situation of applicant	i48	Kind of education of applicant
i35	Max. amt. of supplementary grant		

2.2 Activity Design

In PBWD, the task of activity design is to group the operations in a PDM into activities that form logical pieces of work [8]. As introduced in Sect. 1, activity composition can influence the efficiency of workflow execution [1], as well as the meaningfulness of activities for workflow users [2].

Fig. 2 presents a workflow design for the student grant example based on a proposed set of eight activities. Each activity, depicted in Fig. 3, is designed such that it results in the computation of a data element that directly affects the total amount of student grant an applicant receives, e.g. activity B determines the eligibility of an applicant to receive a grant. Each activity thus represents a significant step during the workflow's execution. The non-atomic activities furthermore encompass distinct sub-processes in the workflow, in which all underlying operations have a similar meaning or subject, e.g. all operations in activity D are related to a student's tuition credit. The proposed design shows that it is possible to partition a PDM into activities that form meaningful tasks in a workflow. The issue of how to design these activities, however, still remains open. Section 3 addresses this by defining objective guidelines for this purpose.

3 Activity Composition

The main contribution in this paper consists of three activity composition guidelines. The guidelines consider the semantic aspects of workflow design. Semantics are those properties that relate to the meaning of PDM elements. These properties are often inferred from the context, rather than explicitly defined in a PDM. Despite this lack of explicitness, it is here proposed that such semantics can be partly derived from the structure of a PDM. This section introduces design

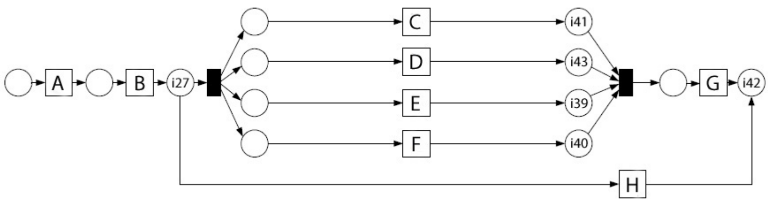

Fig. 2. Process model based on the proposed activity design

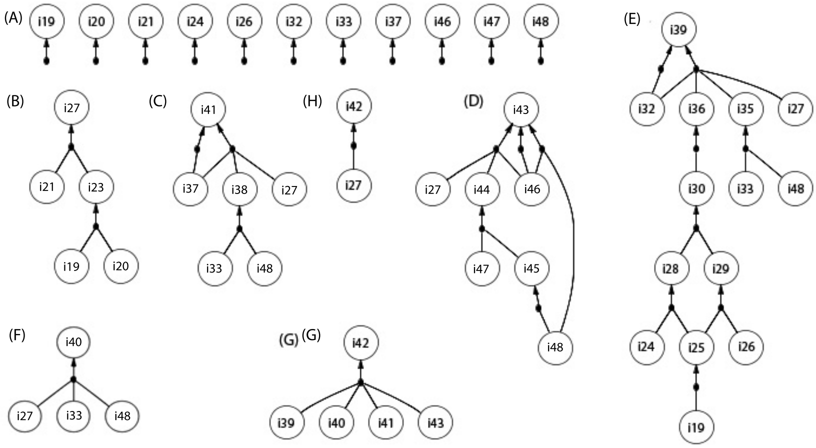

Fig. 3. Non-atomic activities in the process model

guidelines that exploit the proposed relation between semantics and structure. The guidelines are purposefully designed to be independent of context, or any data beyond the structure of a PDM. The proposed guidelines result in activities that work towards the production of an important data element on the one hand and of which the operations are semantically related on the other. This concept is, amongst others, based on the notions of *task identity* and *task significance*, two determinants of *experienced meaningfulness* in the widely used Job Characteristics Model of Hackman & Oldman [2].

This section comprises two parts. Sect. 3.1 considers patterns that imply relative importance of data elements. Secondly, Sect. 3.2 regards the identification of semantically related operations based on structural properties. The guidelines are motivated by referring to the student grants example. However, the propositions are designed to apply to PDMs and PDM-like structures in general.

3.1 Data Element Importance

All data elements in a PDM are to some degree required to produce a value for the root data element. Nonetheless, it is clear that not all data elements are equally important. For example, in the students grants case, data element $i27$, an applicant's right to receive a grant, is certainly more important than $i28$, the income of the applicant's father. The semantic importance of $i27$ is clear, because it fully determines acceptance or rejection. This section proposes that different types of semantically important data elements (IDEs) can be identified based on five structural patterns. The proposed patterns have been found by analysing manual activity designs in [7,8,9] and through utilization of best modelling practices.

Pattern 1 (Root Data). *The root data element is the single data element that is not used as input for any operations.*

Pattern 1 identifies the first type of IDE, namely the root element of a PDM. This element is straightforwardly the most important data element in a PDM, as it represents the final outcome of a workflow. For example in the student grants case, this is the total amount of student grant assigned to an applicant ($i42$).

Pattern 2 (Leaf Data). *A leaf data element is a data element that is produced by an operation without input elements.*

Leaf elements represent the second type of IDEs. These are the data elements that are provided as input to a workflow; the values for these elements are retrieved from outside the process. In the student grants example, the leaf elements represent the data that is retrieved from a student's application, e.g. $i19$, $i24$ and $i26$.

Pattern 3 (Conditional Data). *A conditional data element is a data element that can be produced by multiple alternative operations.*

Pattern 3 identifies *conditional data elements*. These data elements can be produced by multiple alternative operations. Four such elements exist in the student grants example: the root element $i42$, and the data elements $i39$, $i41$, and $i43$. The latter data elements represent three out of the four types of grant that applicants may be eligible to receive. These grants directly affect the value of the root element and are therefore clearly important in this process.

Pattern 4 (Equal Level Data). *An equal level data element is an input data element to an operation that also requires conditional data as input.*

By considering conditional data, three out of the four types of grant are identified as goals. The fourth type of grant is the amount of basic grant assigned to an applicant ($i40$). Without additional context information, there is no reason why this fourth grant is less important than the other types of grant; $i40$ is hence also considered to be of importance. In this case, an important element is thus identified based on its adjacent elements. It is proposed that this transitive notion can be applied in a generic fashion to uncover the fourth type of IDEs: *equal level data.* These elements are revealed by considering the operations that require conditional data as input elements, as defined in Pattern 4.

Pattern 5 (Reference Data). *A reference data element is a data element that is an input element to multiple operations that are, directly or indirectly, involved in the computation of different important data elements.*

Recall that $i27$, the eligibility to receive a grant, is arguably an important data element. $i27$ is important from a structural perspective, because the data element is required to produce multiple other IDEs. Such data elements shall be referred to as *reference data elements*, the final type of IDE.

The five types of IDEs represent those structural patterns that are proposed to predict the significance of data elements. This results in Proposition 1.

Proposition 1 (Important Data Elements). *Root, leaf, conditional, equal level and reference data elements represent important data elements in a PDM.*

3.2 Semantic Relatedness

Semantic relatedness considers the degree to which the meaning of elements is similar. It is here proposed that operations with similar meaning can, to a certain extent, be identified based on structural properties of a PDM. The underlying intuition is that each operation can be associated with the computation of a single IDE. Operations that are associated with the same goal are then considered to be semantically related. Definition 1 defines how operations are associated with a unique IDE[3].

Definition 1 (Associated Element). *The associated element of an operation is the unique IDE for which there exists a path in the PDM from the operation to that IDE, such that this path does not contain any other IDEs.*

Since operations that are associated to the same element are considered to be semantically related, it is proposed that such operations can be grouped into a semantically coherent activity. This proposition is extended with the notion that leaf operations can be grouped together into *leaf activities*. This notion is based on the premise that the values for multiple leaf elements are often retrieved from the same data source. For example, as seen in Sect. 2.1, all leaf elements in the student grants case are derived from a student's application. By grouping leaf operations, the workflow design enforces the retrieval of multiple data elements at once, which is often desirable [11].

Proposition 2 (Semantically Coherent Activities). *A semantically coherent activity is an activity that consists of a set of operations that are associated with the same IDE.*

Proposition 1 and Proposition 2 together form the means to compose activities that result in activities that work towards a goal and of which the operations are semantically related. Finally, it is proposed that an activity composition that conforms to both aforementioned propositions results in well-designed activities.

[3] Due to the way the IDEs, especially reference data elements, are defined in Sect. 3.1, each operation is associated with exactly one IDE. [10] provides a formal proof for this.

Proposition 3 (Well-Designed Activities). *The set of operations in an activity should work towards the production of a relatively important data element and be semantically related to each other.*

4 Conclusions

This paper introduces fundamental guidelines for the objective composition of activities in workflow settings. The proposed guidelines pose that activities should work towards a goal, and should consist of semantically related operations. It has been shown that these properties can be identified based on structural data-flow relations. Hence, the guidelines can be applied objectively.

While this paper emphasizes the motivation for the activity composition problem as well as the development of guidelines, follow-up work has been carried out that does not fit the constraints of this paper. Specifically, an automation of the task of composing activities has been undertaken and a thorough validation of the guidelines has taken place. We intend to report on these in a future publication.

References

1. Seidmann, A., Sundararajan, A.: The effects of task and information asymmetry on business process redesign. International Journal of Production Economics 50(2), 117–128 (1997)
2. Hackman, J.R., Oldham, G.R.: Motivation through the design of work: Test of a theory. Organizational Behavior and Human Performance 16(2), 250–279 (1976)
3. Sun, S.X., Zhao, J.L., Nunamaker, J.F., Sheng, O.R.L.: Formulating the data-flow perspective for business process management. Information Systems Research 17(4), 374–391 (2006)
4. Reijers, H.A., Limam, S., Van der Aalst, W.M.P.: Product-based workflow design. Journal of Management Information Systems 20(1), 229–262 (2003)
5. Trčka, N., van der Aalst, W.M.P., Sidorova, N.: Data-flow anti-patterns: Discovering data-flow errors in workflows. In: van Eck, P., Gordijn, J., Wieringa, R. (eds.) CAiSE 2009. LNCS, vol. 5565, pp. 425–439. Springer, Heidelberg (2009)
6. Robbins, S.P., Judge, T.A., et al.: Organizational behaviour. Pearson Education, Inc., New Jersey (2007)
7. Vanderfeesten, I., Reijers, H.A., Van der Aalst, W.M.P.: Evaluating workflow process designs using cohesion and coupling metrics. Comput. Ind. 59(5), 420 (2008)
8. Vanderfeesten, I.: Product-Based Design and Support of Workflow Processes. PhD thesis, Eindhoven University of Technology (2009)
9. Reijers, H.A.: Design and control of workflow processes: business process management for the service industry. Springer-Verlag (2003)
10. Van der Aa, J.H.: Composing workflow activities. Master's thesis, Eindhoven University of Technology (2013)
11. Reijers, H.A., Mansar, S.L.: Best practices in business process redesign: an overview and qualitative evaluation of successful redesign heuristics. Omega 33(4), 283–306 (2005)