

Evaluating workflow process designs using cohesion and coupling metrics

Irene Vanderfeesten^{*}, Hajo A. Reijers, Wil M.P. van der Aalst

Technische Universiteit Eindhoven, Department of Technology Management, PO Box 513, NL-5600 MB Eindhoven, The Netherlands

Received 9 January 2007; received in revised form 5 December 2007; accepted 13 December 2007

Available online 4 March 2008

Abstract

Building on the similarities between software programs and workflow processes, this paper proposes a heuristic that offers guidance for the creation and evaluation of process designs in administrative settings. Designers can use this heuristic to select from several alternatives the process design that is strongly cohesive and weakly coupled. It is argued that such a design will result in fewer errors during information exchanges and in more understandable activity descriptions. The paper includes an application of the heuristic in an industrial workflow setting, which supports its feasibility and practical value. The paper also presents the freely available *CoCoFlow* tool that implements the heuristic and its associated metrics. © 2007 Elsevier B.V. All rights reserved.

Keywords: Business process modelling; Workflow management; Quality metrics

1. Introduction

In 2006, one of the world's largest asset managing companies redesigned its business process to produce the annual reports for its investment funds. On the basis of a meticulous analysis of the essential information processing function of the process, a new process design was developed using the method of Product-Based Workflow Design (PBWD) [45]. The new design reshuffles the order of the original steps and exploits the opportunities of an automated, XML-based message exchange between investment managers, accountants, print shop, etc. It is anticipated that this new way of working will cut the cycle time of the process by half. Not only does this imply a better service to the company's customers, e.g. corporate investors and pension funds, but also it facilitates compliance with deadlines from the financial authorities.

This case illustrates that the combination of *process redesign* and the *application of IT* are viable today, even though Business Process Reengineering (BPR) was introduced over a decade ago [25]. Particularly in administrative settings, where the focus is on information processing (e.g. the evaluation of a damage claim, the issuing of a building permit, or the handling

of a mortgage application), various redesign opportunities exist [39]. For example, by using electronic documents and workflow technology it is relatively easy to change the routing of a file along the various decision-making steps in a process. When doing this properly, the average execution cost of the overall process can be minimized [3].

However, coping with such design freedom may also be problematic. The main issue that we focus on in this paper is the proper size of the individual *activities* (or *tasks*) in a process. This design choice is known as the issue of *process granularity* [18]. Badly chosen sizes of activities in a process may negatively affect its performance when being executed or enlarge the maintenance burden of the process model in case of updates. Small activities, on the one hand, may increase the number of hand-offs between activities leading to an increase of errors [42,49]. Large activities, on the other hand, may become unworkable to be executed well by humans [18,42].

This paper addresses the problem of activity design in the domain of information-intensive processes, typically found in the service industry. This issue is particularly relevant within the setting of BPR projects [26] where also the application is being considered of workflow management technology [5]. In the remainder of this paper, we shall refer to such processes as *workflow processes* or simply as *workflows*. We will present *cohesion* and *coupling metrics* that focus on the content of activities, i.e. their *operations*. By using the proposed set of metrics, it can be quantitatively expressed to what extent operations “belong” to each other within one activity or, in

^{*} Corresponding author. Tel.: +31 40 247 4366; fax: +31 40 243 2612.

E-mail addresses: i.t.p.vanderfeesten@tue.nl (I. Vanderfeesten),
h.a.reijers@tue.nl (H.A. Reijers), w.m.p.v.d.aalst@tue.nl
(W.M.P. van der Aalst).

other words, how *cohesive* such an activity is. In addition, it is important to measure to what extent the various activities are dependent on each other or, in other words, how much they are *coupled*. The inspiration for the proposed metrics comes from software engineering, where an old design aphorism is to strive for *strong cohesion, and loose coupling*.

We build on work by Selby and Basili [50] and Xenos et al. [59], who defined coupling and cohesion metrics for software programs. The proposed solution based on comparable metrics fits conceptually with the PBWD method, as applied in the case we mentioned in the opening. PBWD is a BPR method which is adopted and actively applied by one of the Big Four consultancy firms [45]. For a more detailed application of the method in the financial services, see ref. [41].

In this paper, findings from our earlier work [43,46] are integrated and extended in several directions. A first version of the workflow cohesion metric was introduced in ref. [43], but it lacked some facilities. For example, it could not handle conditional alternatives to achieve the same output (XOR-construct), which is a construct very common in workflow processes [6]. Secondly, we introduced in ref. [46] an additional coupling metric, but the integrated design heuristic did not yet consider resource issues that in real life impose constraints on good designs. Thirdly, in our experiments we noticed that the manual application of the metrics can be time-consuming and may easily lead to human errors. Therefore, this paper introduces the freely available *CoCoFlow* tool that can be used for the automatic computation of the metrics. Finally, we applied the integrated design heuristic to the case of a Dutch governmental agency to demonstrate its feasibility and practical value.

The structure of this paper is as follows. In Section 2, we introduce a motivating example that clarifies the goal of our research. Section 3 describes the cohesion and coupling metrics and the heuristic for selecting a favorable design among various alternatives. The fourth section contains an application in an industrial setting, which is followed by a description of the *CoCoFlow* tool in Section 5. In Section 6, we elaborate on related work in the fields of software engineering and workflow to position our contribution. The paper ends with a discussion and prospects for future work.

2. Motivating example

To motivate the application of cohesion and coupling metrics on the one hand and to introduce some relevant concepts on the other, we present in this section a workflow 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 Informatie Beheer Groep¹(IBG) under the authority of the Dutch Ministry of Education, Culture and Science. We will first present the essential information processing function that the workflow process must perform. Then, we will illustrate the

design dilemma that we focus on in this paper, by representing three alternative process designs.

2.1. An information processing perspective

As in many other workflow processes, the essential output of the IBG workflow process can be clearly distinguished. In this case, it is the calculation of the government grant that any new student can apply for. The proper amount of money to be granted must be determined by taking several factors into account: the student's background, the income of the student's parents, the student's living situation, the kind of study grant the student applied for, etc. Each of these factors in turn may rely on other factors.

We shall refer to the various relevant factors as *information elements*. In the activities of a workflow process the values of information elements are used to produce new values for other information elements. In the end, the essential output is produced as a set of information element values.²

An activity in a workflow process will now be considered as a number of *operations* on information elements. Each operation has one or more *input* information elements and one or more *output* information elements. An operation is an atomic processing step, so it cannot have “half-assembly” products, i.e. intermediate information element values. A workflow activity, on the contrary, is composed of one or more operations. Therefore, it can have “half-assembly” products. The problem we raised in Section 1 can now be reformulated as *the proper clustering into activities of operations on information elements*. In Fig. 1, the complete *information element structure* of the IBG process can be found. At the top, information element 42 can be distinguished, which represents the total grant amount that is decided to be assigned to the applicant. (Note that a complete description of the individual information elements is provided in Appendix A (see Table A.1).) The information element structure expresses that a value for information element 42 can be determined on the basis of values for information elements 39, 40, and 41, which respectively represent the granted amount of supplementary scholarship, basic scholarship, and the loan component. This relation between information elements 39, 40, 41, and 42 is expressed by arrows leading from the input elements, “knotted” together, to the output element. The operation to be performed on the input elements in this case is to simply add their values.³

Considering the information element structure for this case again (Fig. 1), it is interesting to note that a value for information element 42 can also be computed in a different way. This different computation is based on the value of information element 27 alone, which represents the outcome of the decision whether the respective student may receive a government grant at all. If this value is determined to be “no” – which is the case if the student is older than thirty or does not

¹ <http://www.ib-groep.nl>.

² Typically, one value is of prime interest.

³ Note that the content of operations is not expressed in this figure.

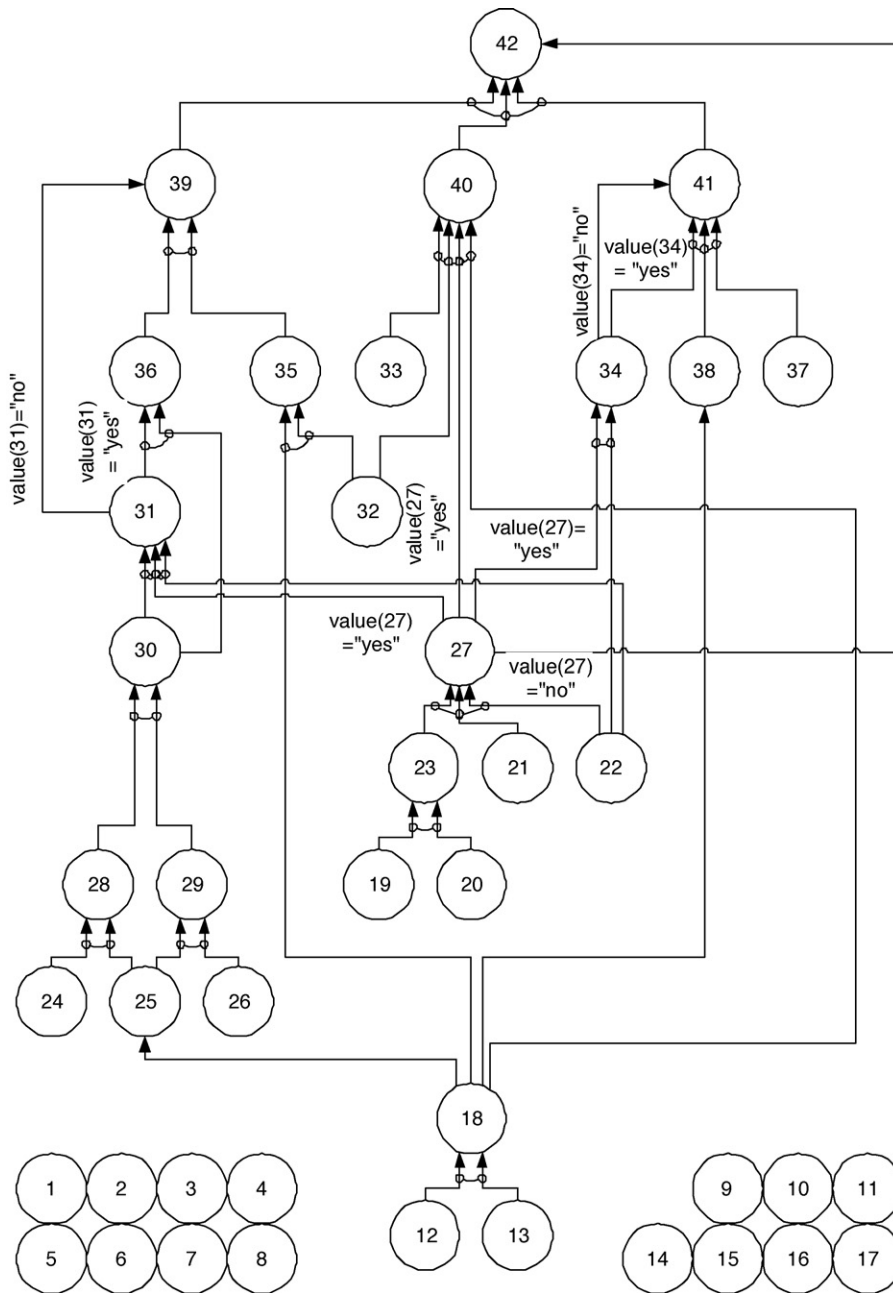


Fig. 1. Information element structure for the process “Handling of requests for governmental student grants”.

have the Dutch nationality – the value of information element 42 can be produced directly (leading to a zero grant). When the value of information element 27 equals “yes”, a number of other steps have to be executed before the value of information element 42 can be properly determined. As the reader can observe in Fig. 1, some of the information elements are no output of any operation, e.g. information element 37 has no incoming arcs. Such an information element does not require a computation or decision within the process, but its value is obtained from other sources. In this case, it represents the size of the loan that the applicant requests for as part of the grant. It is therefore provided by the applicant himself/herself. Also, some information elements are not integrated in any of the operations, i.e. information elements 1–11 and 14–17. The kind

of information they represent is typically required for proper identification, registration, and communication with various parties, but plays no important role in the workflow’s information processing. For completeness’ sake, they have been included in the example but they are left out in further explanations. Finally, note that the information element structure is a *network* structure (not a tree structure) and that it is required to be *connected* and *acyclic*.

The use of an information element structure to capture the essential information function of a workflow process is very similar to the use of a Bill Of Material (BOM) [38] for the design of assembly lines in manufacturing. The information input–output perspective on workflow processes is taken from the methodology of *Product-Based Workflow Design* [2,42,45],

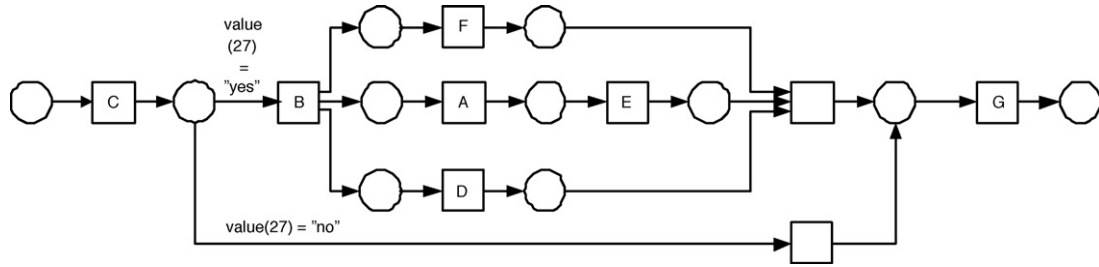


Fig. 2. Process model of the first alternative.

which has been successfully applied in practice to analyze and redesign workflow processes in banking and social services domains. Also, some contemporary Workflow Management Systems adopt a comparable information-centered modelling and execution approach [4].

2.2. Three alternative designs

Let us now consider a proper design of a workflow process that (i) includes all information elements and operations from the IBG information element structure, (ii) respects their dependencies. Another important aspect of a proper workflow design would be that only operations that require similar skills or authorization levels should be grouped together within the same activity. Although we capture this requirement in our formalization of the design problem later in this paper, we abstract from it in this motivating example.

In Fig. 2, a preliminary design is presented that implements the information structure of Fig. 1. The design is represented as a workflow net [5]. The seven individual activities are represented as labelled rectangles, which contribute to the

processing as follows. First, in activity C it is determined whether the applicant is entitled to a governmental grant at all. If so, the period/year of student grant and a reference year are determined in activity B. In the parallel activities F, A and D, the loan amount, the income of the applicant’s parents, and the amount of basic scholarship are determined respectively. Activity E follows up on activity A by determining the amount of supplementary scholarship. Finally, in activity G the total grant amount can be determined (which is zero if activity C leads to a “knock-out”).

Fig. 3 shows how this design exactly partitions the information element structure over the various activities. For example, activity G incorporates both operations to determine a value for information element 42. The structure of the process model ensures that the correct information element values have been determined before activity G can be executed at all. As an alternative to this design, a model can be considered where activities A and E are split up into four new activities: A1, A2, A3, and A4. In this alternative design, the activities are considerably smaller. For example, the operations to determine values for respectively information elements 28 and 29 used to

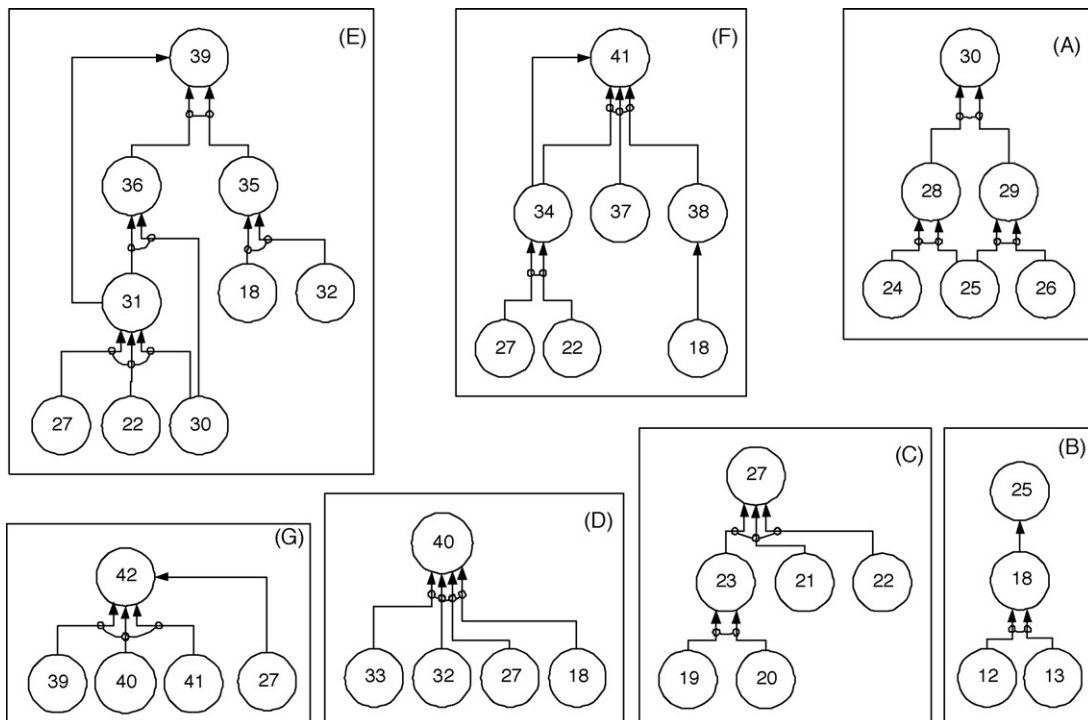


Fig. 3. The partitioning of the information element structure for the first alternative design.

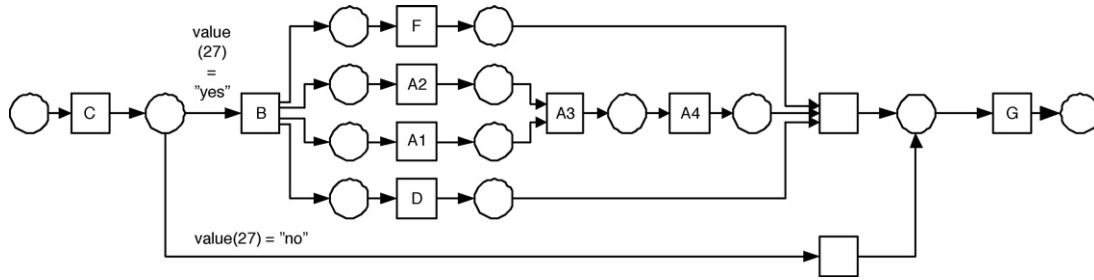


Fig. 4. Process model of the second alternative. The original activities A and E are split up.

be clustered within activity A, but here become separate activities (A1 and A2). Still, the overall design conforms to the information element structure (see Figs. 4 and 5). The total number of activities in this alternative is nine.

Yet another alternative would be to merge activities A and E from the original design together (see Figs. 6 and 7). This proper process design has six, somewhat larger activities.

Looking at the three proposed designs, the design dilemma of interest is: “Which design is preferable?”. Each of the presented alternatives is a correct translation of the information

element structure, as they cover all relevant information elements and operations and respect their dependencies. The alternatives only differ in the choices with respect to the distribution of operations over activities.

Earlier experiments we carried out with a group of experienced workflow designers suggest that they would favor the first process design, because it incorporates activities which are neither too large nor too small [43]. In ref. [18] similar guidance can be found (see Section 6). As will be shown in the remainder of this paper, the proposed heuristic

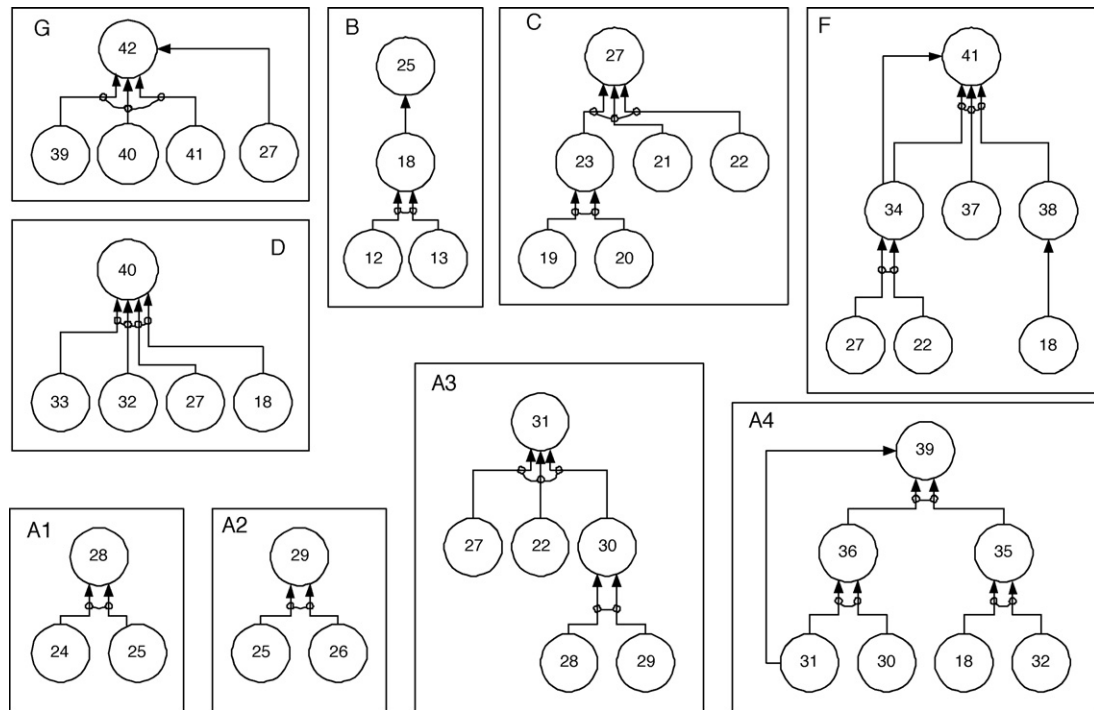


Fig. 5. The partitioning of the information element structure in smaller activities. The original activities A and E are split up.

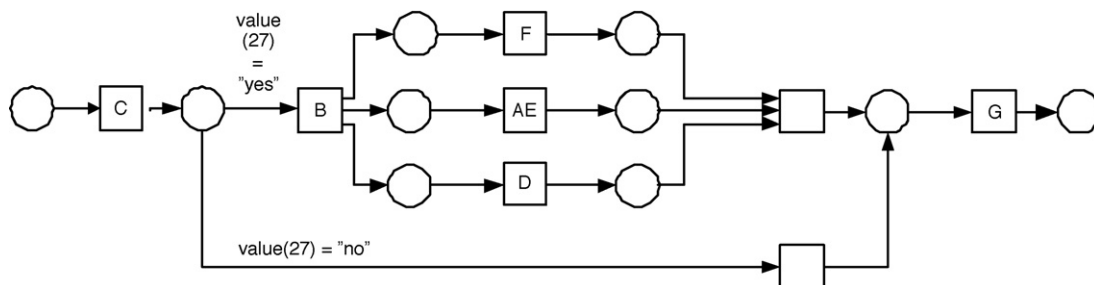


Fig. 6. Process model of the third alternative. The original activities A and E are combined.

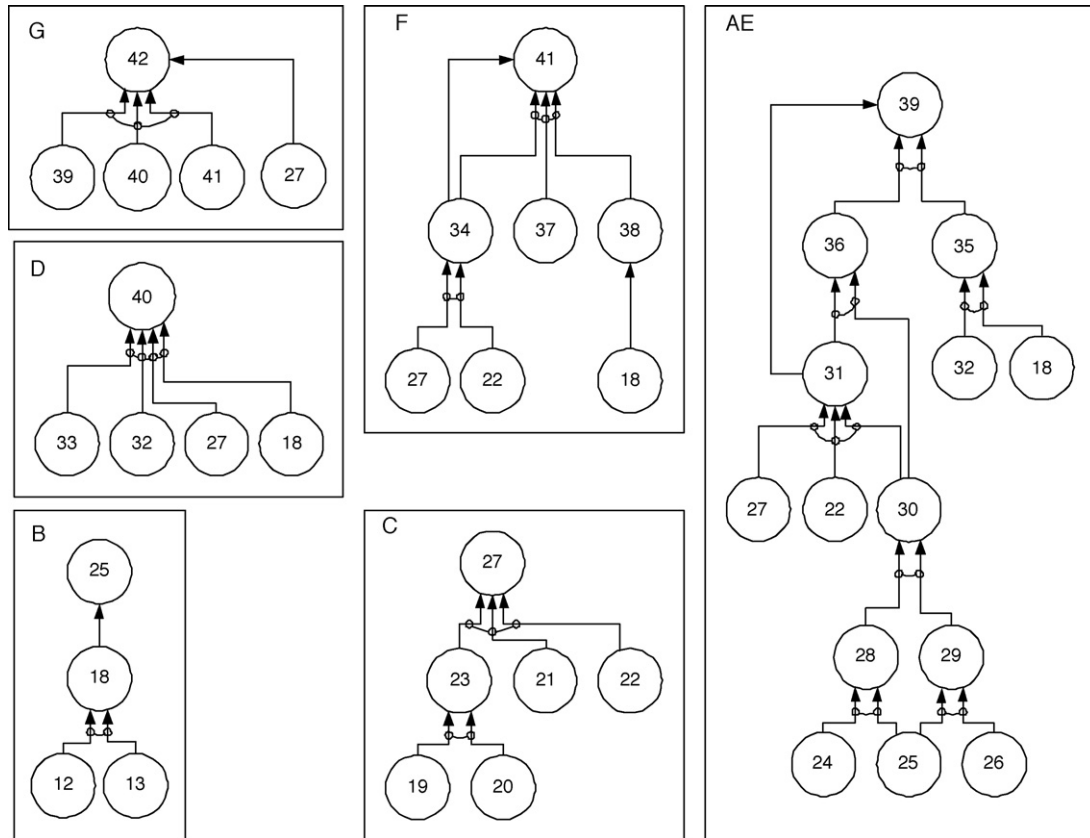


Fig. 7. The partitioning of the information element structure in larger activities. The original activities A and E are combined.

will offer concrete guidance for dealing with this design dilemma.

3. Cohesion and coupling metrics

To argue for the viability of using coupling and cohesion metrics for workflow design, it is important to note that software programs and workflow processes are quite similar in some respects. We distinguish the following similarities:

- They both focus on *information processing*. Within each step, one or more outputs are produced on the basis of one or more inputs.
- They have a similar *compositional structure*. A program—functional or object-oriented—can be split up into modules or classes. Every module consists of a number of statements, and every statement contains a number of variables and constants. Likewise, a workflow process has activities. Each activity is composed of elementary operations and each operation uses one or more pieces of information to produce new information (Table 1).⁴
- Their *dynamic execution* is derived from a static structure. When instantiating either a software program or a workflow

Table 1
Similarities between software programs and workflow processes

Software programs	Workflow processes
Program	Workflow process
Module/class	Activity
Statement/method	Operation
Variable/constant	Data element

process, an execution flow of their elements takes place in accordance with their static representation. This flow may involve consecutive executions, concurrency, conditional routings, etc.

Because of such similarities – which have been noted before, e.g. in ref. [24]– it is conceivable that metrics used in software engineering are applicable in the domain of workflow processes as well. In software engineering, manipulations (declarations, assignments, invocations, etc.) that are strongly related are preferably grouped together within the same module or class [53]. There are clear implications for a better maintainability of programs by using this approach and there is also considerable empirical evidence that the resulting computer programs contain fewer run-time errors [12,50]. Yet, none of the existing software engineering metrics we studied seem to be directly applicable to workflow processes [46].

By adapting and translating quality metrics from the software engineering to the workflow domain, we expect to

⁴ Note that the resources executing a program or process are different; the steps of a software programs are run automatically on a computer, while the activities of a business process often are performed by humans.

offer concrete guidance for improving upon the following workflow model properties:

- *Execution quality*, which can be measured by the number of errors being made during execution of the captured process or by the degree to which accomplished work meets common standards.
- *Maintainability*, which can be defined as the ease with which a workflow process model can be updated because it is easily understood.

Based on the strong similarities between software programs and workflow processes, we consider *high cohesion and loose coupling* as a valuable design maxim in the workflow domain as well. A *loose coupling* of activities will result in few information elements that need to be exchanged between activities in a workflow process, reducing the probability of run-time mistakes. *Highly cohesive* activities, in turn, are likely to be understood and performed better by people than large chunks of unrelated work being grouped together. Because the creation of large activities will decrease the degree of coupling and the creation of small activities will increase cohesion, it seems that high cohesion and loose coupling is providing the right balance to improve upon both execution quality and maintainability of a workflow model.

3.1. Formalization

The main objective in the formalization of cohesion and coupling metrics is to support the job of *workflow process design*. This job can be understood as imposing on a set of operations a number of activities that partition that set (cf. the grouping of operations into activities for the three alternative designs of our motivating example, Section 2.2). First the structure of information processing is formalized using a concept that is called the *operations structure*.

Definition 1 (Operations structure).

An operations structure is a tuple (D, W, O) with:

- D : the set of *information elements* that are being processed.
- W : the set of *resource classes* or *roles* that are available to the process. A relation \preceq is defined on these resource classes.⁵ ‘ $v \preceq w$ ’ means that a person with role w is allowed to do all the work v is allowed to do (and potentially more).⁶
- $O \subseteq D \times W \times \mathcal{P}(D)$: the set of *operations* on the information elements, such that there are no “dangling” information elements and no value of an information element depends on itself, i.e. the graph (V, E) with $V = D$ and $E = \{(p, c) \in D \times D \mid \exists (p, w, cs) \in O (c \in cs)\}$ is connected and acyclic.

⁵ Note that the \preceq -relation is a partial order, i.e. it is transitive and reflexive, but not symmetric.

⁶ For instance, if the head of a financial department is allowed to do the jobs of his accountant and of his secretary (besides his own jobs involved with being the head of the department), it can be formalized as follows: *accountant* \preceq *head of financial department*, *secretary* \preceq *head of financial department*.

So, if operation $(p, w, cs) \in O$ for a given operations structure (D, W, O) , this means that it is possible for a resource with role w to produce a value for information element p on the basis of values for the set of information elements cs . For convenience, we introduce the following notations as shorthand: $\bar{O} = \{(p, cs) \mid \exists (p, w, cs) \in O\}$, for operations without reference to resources, and $\hat{O} = \{(p, c) \mid \exists (p, w, cs) \in O (c \in cs)\}$, for relations between information elements.

Let us consider the IBG process introduced in Section 2 again, in particular the information element structure of Fig. 1. We assume that there is just a single resource type w of relevance within this context, so $W = \{w\}$. The operations structure for the IBG process is then defined as follows:

- $D = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42\}$
- $W = \{w\}$
- $O = \{(18, w, \{12, 13\}), (23, w, \{19, 20\}), (25, w, \{18\}), (27, w, \{21, 22, 23\}), (28, w, \{24, 25\}), (29, w, \{25, 26\}), (30, w, \{28, 29\}), (31, w, \{22, 27, 30\}), (34, w, \{22, 27\}), (35, w, \{18, 32\}), (36, w, \{30, 31\}), (38, w, \{18\}), (39, w, \{35, 36\}), (39, w, \{31\}), (40, w, \{18, 27, 32, 33\}), (41, w, \{34, 37, 38\}), (41, w, \{34\}), (42, w, \{39, 40, 41\}), (42, w, \{27\})\}$
- $\bar{O} = \{(18, \{12, 13\}), (23, \{19, 20\}), (25, \{18\}), (27, \{21, 22, 23\}), (28, \{24, 25\}), (29, \{25, 26\}), (30, \{28, 29\}), (31, \{22, 27, 30\}), (34, \{22, 27\}), (35, \{18, 32\}), (36, \{30, 31\}), (38, \{18\}), (39, \{35, 36\}), (39, \{31\}), (40, \{18, 27, 32, 33\}), (41, \{34, 37, 38\}), (41, \{34\}), (42, \{39, 40, 41\}), (42, \{27\})\}$
- $\hat{O} = \{(18, 12), (18, 13), (23, 19), (23, 20), (25, 18), (27, 21), (27, 22), (27, 23), (28, 24), (28, 25), (29, 25), (29, 26), (30, 28), (30, 29), (31, 22), (31, 27), (31, 30), (34, 22), (34, 27), (35, 18), (35, 32), (36, 30), (36, 31), (38, 18), (39, 35), (39, 36), (39, 31), (40, 18), (40, 27), (40, 32), (40, 33), (41, 34), (41, 37), (41, 38), (41, 34), (42, 39), (42, 40), (42, 41), (42, 27)\}$

The set of information elements D for the IBG operations structure for example includes elements 27, 39, 40, 41, and 42. The set of operations O also includes $(42, w, \{39, 40, 41\})$ and $(42, w, \{27\})$, expressing that a value for information element 42 can be determined by a resource of type w on basis of either the values of elements 39, 40, and 41 or on the basis of 27 alone. Note that for most other information elements just a single operation is available to determine its value.

The operations structure (D, W, O) can be considered as the starting point for a workflow process design, capturing the essential information processing that has to be performed. To discuss how the activities of a design may properly partition

such an operations structure, we define the notion of an *activity* and a *process* on the operations structure.

Definition 2 (Activity).

An *activity* T on operations structure (D, W, O) is a tuple $(t, e) \in \mathcal{P}(O) \times W$ with

- t : a set of operations $(t = \{(p_1, w_1, cs_1), (p_2, w_2, cs_2), \dots\})$, and
- e : the resource that is allowed to execute the activity, fulfilling the following requirement:

$$\forall_{(p,w,cs) \in t} (w \preceq e).$$

As a shorthand we introduce the following notations, assuming $T = (t, e)$:⁷ $\bar{T} = \{(p, cs) \mid \exists_{(p,w,cs) \in t}\}$, for the operations within an activity without references to resources, $\hat{T} = \{(p, c) \mid \exists_{(p,w,cs) \in t} (c \in cs)\}$, for the input–output relations in an activity, and $\tilde{T} = \bigcup_{(p,c) \in \hat{T}} \{p, c\}$, for the information elements processed in an activity.

In essence, any group of operations may be clustered together into an activity, as long as a resource type can be distinguished within the available ones that is capable of performing all operations. After all, only fully qualified and authorized employees may execute a workflow activity (see, e.g. [5]).

In the IBG process all operations and activities have the same resource for reasons of understandability. Thus, we could for instance define an activity $T = (t, w)$ consisting of operations $(31, w, \{22, 27, 30\})$ and $(36, w, \{30, 31\})$. The accompanying notations have the following elements for this example:

$$\begin{aligned} \bar{T} &= \{(31, \{22, 27, 30\}), (36, \{30, 31\})\} \\ \hat{T} &= \{(31, 22), (31, 27), (31, 30), (36, 30), (36, 31)\} \\ \tilde{T} &= \{22, 27, 30, 31, 36\} \end{aligned}$$

Note that the proposed activity T is not part of one of the alternative designs of the motivating example.

At this point it becomes important how the various activities together constitute a workflow process. Several issues emerge when considering this, most importantly (i) how the *control flow* is to be specified and (ii) which *correctness* criteria we consider relevant. With respect to the control flow, many different process modelling languages are available and, on a more abstract level, several routing constructs exist [6]. With respect to correctness, an important criterion that comes to mind is, e.g. *soundness* [1], which ensures completion and proper termination of a workflow process. Considering the operations that are used in activities, it also seems reasonable to require some guarantee that information is indeed available when it is to be used as input to calculate new information.

However, for the purpose of this paper these issues are less relevant, as they do not affect the computation of the various metrics. We will therefore restrict ourselves to simply considering a process as a set of activities. For a more formal treatment of this subject the reader is referred to [42].

Note that for all examples in this paper holds that workflow nets are used as modelling technique and only simple sequential and choice constructs are applied (control flow issues). Also, all process models cover all available operations from the information element structure, the models are all sound, and the specified control flow ensures that the value of an information element is determined when it is to be used as an input element (correctness issues). Depending on the specific context and one's preferences, alternative choices for these issues may be considered in other cases.

Definition 3 (Process).

A process S on an operations structure (D, W, O) is a set of activities:

$$S \subseteq \mathcal{P}(O) \times W$$

Again we introduce a simplified notation as a shorthand:

$$\bar{S} = \{t \mid \exists_{(t,e) \in S}\}, \text{ for activities without reference to resources.}$$

The first alternative process in the IBG case contains a set of seven activities: $S = \{A, B, C, D, E, F, G\}$, which is shown in Figs. 2 and 3. The partitioning of the operations over activities A to G is captured as follows:

$$\begin{aligned} O_A &= \{(28, w, \{24, 25\}), (29, w, \{25, 26\}), \\ &\quad (30, w, \{28, 29\})\} \\ O_B &= \{(18, w, \{12, 13\}), (25, w, \{18\})\} \\ O_C &= \{(23, w, \{19, 20\}), (27, w, \{21, 22, 23\})\} \\ O_D &= \{(40, w, \{18, 27, 32, 33\})\} \\ O_E &= \{(31, w, \{22, 27, 30\}), (35, w, \{18, 32\}), \\ &\quad (36, w, \{30, 31\}), (39, w, \{35, 36\}), (39, w, \{31\})\} \\ O_F &= \{(34, w, \{22, 27\}), (38, w, \{18\}), \\ &\quad (41, w, \{34, 37, 38\}), (41, w, \{34\})\} \\ O_G &= \{(42, w, \{39, 40, 41\}), (42, w, \{27\})\} \end{aligned}$$

Building on this simple notion of a process, a *cohesion* metric can now be defined as follows. Its first component, the *relation cohesion*, quantifies how much the different operations within one activity are related. It does so by determining for each operation of an activity with how many other operations it overlaps by sharing an input or output, i.e. a non-empty intersection.

In this determination conditional alternatives are treated as separate operations, because in an instantiation only one of the alternatives will be executed. The overlap between these operations (i.e. the same output element) is therefore not

⁷ Thus, $\bar{T} \subseteq \bar{O}$, $\hat{T} \subseteq \hat{O}$, and $\tilde{T} \subseteq \tilde{O}$.

considered. This explains why the expression $p_1 \neq p_2$ is in the formula.⁸

Next, the average overlap per operation is computed by dividing the total amount of overlaps by the number of operations. Finally, note that all overlaps are counted twice, because we considered all pairs of operations separately (for example distinguishing (p_1, w_1, cs_1) , (p_2, w_2, cs_2) and (p_2, w_2, cs_2) , (p_1, w_1, cs_1) as different pairs). Therefore, to get a relative metric score between 0 and 1, the average overlap per operation over all operations within an activity is divided by the maximal overlap, i.e. the number of operations minus one.

Definition 4 (Activity relation cohesion).

For an activity $T = (t, e)$ on an operations structure (D, W, O) , the activity relation cohesion $\lambda(T)$ is defined as follows:

$$\lambda(T) = \begin{cases} \frac{|\{((p_1, cs_1), (p_2, cs_2)) \in \bar{T} \times \bar{T} | (\{p_1\} \cup cs_1) \cap (\{p_2\} \cup cs_2) \neq \emptyset \wedge p_1 \neq p_2\}|}{|\bar{T}| \cdot (|\bar{T}| - 1)}, & \text{for } |\bar{T}| > 1 \\ 0, & \text{for } |\bar{T}| \leq 1 \end{cases}$$

The other component of our cohesion metric, the *activity information cohesion*, focuses on all information elements that are used either as input or as output by any operation within the respective activity. It determines how many information elements are used more than once in proportion to all the information elements used. It does so by counting all different information elements that appear in the intersection of a pair of operations, considering all pairs. This number is divided by the total number of information elements in the activity, i.e. $(|\hat{T}|)$.

Definition 5 (Activity information cohesion).

For an activity $T = (t, e)$ on an operations structure (D, W, O) , the information cohesion $\mu(T)$ is defined as follows:

$$\mu(T) = \begin{cases} \frac{|\{d \in D | \exists (p_1, cs_1), (p_2, cs_2) \in \hat{T} (d \in (\{p_1\} \cup cs_1) \cap (\{p_2\} \cup cs_2)) \wedge (p_1 \neq p_2))\}|}{|\hat{T}|}, & \text{for } |\hat{T}| > 0 \\ 0, & \text{for } |\hat{T}| = 0 \end{cases}$$

The total cohesion of an activity is now given as the product of both the relation and information cohesion. This is to reflect that in our opinion an activity has to score high on both cohesion aspects to say it is cohesive. In other words, the operations clustered together should (i) be interrelated to each other and (ii) information should be shared to a certain degree.

Definition 6 (Activity cohesion).

For an activity $T = (t, e)$ on an operations structure (D, W, O) , the activity cohesion $c(T)$ is defined as follows:

$$c(T) = \lambda(T) \cdot \mu(T)$$

⁸ Note that overlapping output elements hint at alternative execution paths (i.e. XOR-splits in a process model), while overlapping input elements only point at information sharing.

Getting back to the IBG example, the relation cohesion, the information cohesion and the total activity cohesion of activity E in the first IBG design are:

$$\begin{aligned} \lambda(E) &= \frac{|\{((31, \{22, 27, 30\}), (36, \{30, 31\})), \dots\}|}{5 \times 4} \\ &= \frac{10}{20} = 0.5 \\ \mu(E) &= \frac{|\{30, 31, 35, 36\}|}{9} = \frac{4}{9} \approx 0.444 \\ c(E) &= 0.5 \times \frac{4}{9} \approx 0.222 \end{aligned}$$

The overall cohesion of the workflow process can then be determined by the average activity cohesion.

Definition 7 (Process cohesion).

For a process which consists of a set of activities (S) on the operations structure (D, W, O) , the average cohesion, or process cohesion ch , is defined as follows:

$$ch = \frac{\sum_{t \in \bar{S}} c(t)}{|\bar{S}|}$$

As an extension and a natural counterpart of cohesion we also define a metric for *coupling* in a process. Coupling focuses on how strongly the activities in a workflow process are related, or connected, to each other. A certain activity is connected to another one if and only if they share one or more information elements. The coupling metric determines the number of related activities for each activity. First, the average coupling is determined by adding up the number of connections for all activities and dividing this number by the total number of activities. Now all pairs of activities have been counted twice. To get a relative score for this metric, the average coupling is divided by the maximal number of coupling, i.e. the number of activities minus one.

Definition 8 (Process coupling).

For a process that consists of a set of activities (S) on the operations structure (D, W, O) , the process coupling cp is defined as follows:

$$cp = \begin{cases} \frac{|\{(T_1, T_2) \in S \times S | \bar{T}_1 \neq \bar{T}_2 \wedge (\hat{T}_1 \cap \hat{T}_2) \neq \emptyset\}|}{|\bar{S}| \cdot (|\bar{S}| - 1)}, & \text{for } |S| > 1 \\ 0, & \text{for } |S| \leq 1 \end{cases}$$

Inspired by the work of Selby and Basili [50], we now define a *coupling/cohesion ratio*. This ratio enables the comparison between various design alternatives.

Definition 9 (Process coupling/cohesion ratio).

For a process which consists of a set of activities (S) on an operations structure (D, W, O) , the process coupling/cohesion

ratio ρ is defined as follows:

$$\rho = \frac{cp}{ch}$$

Again we can calculate these metrics for our example. The process cohesion of the IBG design is:

$$ch = \frac{0.5 + 0.25 + 0.167 + 0 + 0.22 + 0.143 + 0}{7} \approx 0.183$$

In the IBG process design activity E is for example coupled with 6 other activities. The total process coupling of the IBG design is:

$$cp = \frac{|{(A, B), (A, E), (B, A), \dots, (G, E), (G, F)}|}{7 \times 6}$$

$$= \frac{30}{42} \approx 0.714$$

For the first alternative design in the IBG case the coupling/cohesion ratio is:

$$\rho = \frac{0.714}{0.183} \approx 3.9$$

The previously defined metrics can be used to find the preferable workflow design among a number of alternative designs, considering its execution quality and maintainability. The design with the minimal process coupling/cohesion ratio is the most favorable design. Note that we do not describe how the alternative designs can be determined. The ratio can only help to choose the best option from already specified alternatives.

In case of our motivating example the coupling/cohesion ratios for the three alternative designs are 3.9, 5.8, and 7.0, respectively. Therefore, the first design is preferable, which seems to be confirmed by our intuition: It contains activities which are neither too large nor too small.

4. Application

To show the feasibility and practical relevance of the previously defined metrics, we will study an industrial case in this section. The subject is the governmental UWV agency⁹ (formerly known as GAK) which is responsible for awarding unemployment benefits in the Netherlands. The process in question deals with the decision that the UWV has to make on whether or not to grant such benefits once a request has been received. Typically, factors should be taken into account such as the reason for the applicant to have become unemployed, the length of the period that the previous job was held, etc.

In our previous research, a design for this particular process was derived using the PBWD method we mentioned before. For more detailed information on this case and the derivation of the design, the reader is referred to ref. [42,45].

In this section we will reconsider this design primarily from an activity design perspective. Note that this issue was not considered in much depth in the original redesign project, where the main driver was to look for a way to minimize the average effort per case in terms of human labor hours.

Fig. 8 displays the information element structure for the UWV case. The resulting workflow design can be seen in Fig. 9. Originally, the design was derived taking into account several requirements and restrictions. For example, the processing order of the various operations was determined such that the expected number of additional work at any point in the process was minimized for the average case. In this application, it turned out to be wise to check the conditions first that would most likely lead to a knock-out [3], i.e. a decision not to grant any unemployment benefits. Also, the management of the UWV found it very important to retrieve all information from the client directly at the start of the process. In this way, there would never be any need for contacting the client again during process execution, avoiding long delays. This is one of the so-called “best practices” that can often be distinguished in redesign projects, as discussed in more detail in ref. [44]. Together, these considerations motivate the exact ordering of activities of the workflow design in question.

Eventually, after the proper processing order had been established, the definition of the activities was determined in an intuitive way, leading to a rather fragmented design. On the basis of the original process design, two alternatives are now considered. Alternative 1 is a slight modification of the original design. Because the original design contains a number of activities that only contain a single operation, i.e. activities C, D, E, and F, in Fig. 9, these were combined into one larger activity. Alternative 2 takes this approach one step further by not only joining activities C–F, but also by merging the activities that (partly) contain the same operations (i.e. activities J and M). Overall, both alternatives aim to counter the fragmentation of the original design.

Table 2 shows the values for cohesion and coupling metrics for the original design and the two alternatives. Based on a desirable low value for the coupling/cohesion ratio, alternative 2 is the best option. Considering this alternative, it can be noticed that it does not contain unnecessary small or superfluous activities. This appeals to our intuition that work should not become too fragmented (see Section 3). As a result, we expect this alternative design to be the one that is more

Table 2
Cohesion and coupling for five alternative designs to the UWV case

	Process cohesion	Process coupling	Coupling/cohesion ratio
Original design	0.076	0.788	10.3
Alternative 1	0.102	0.806	7.9
Alternative 2	0.116	0.810	7.0

⁹ <http://www.uwv.nl>.

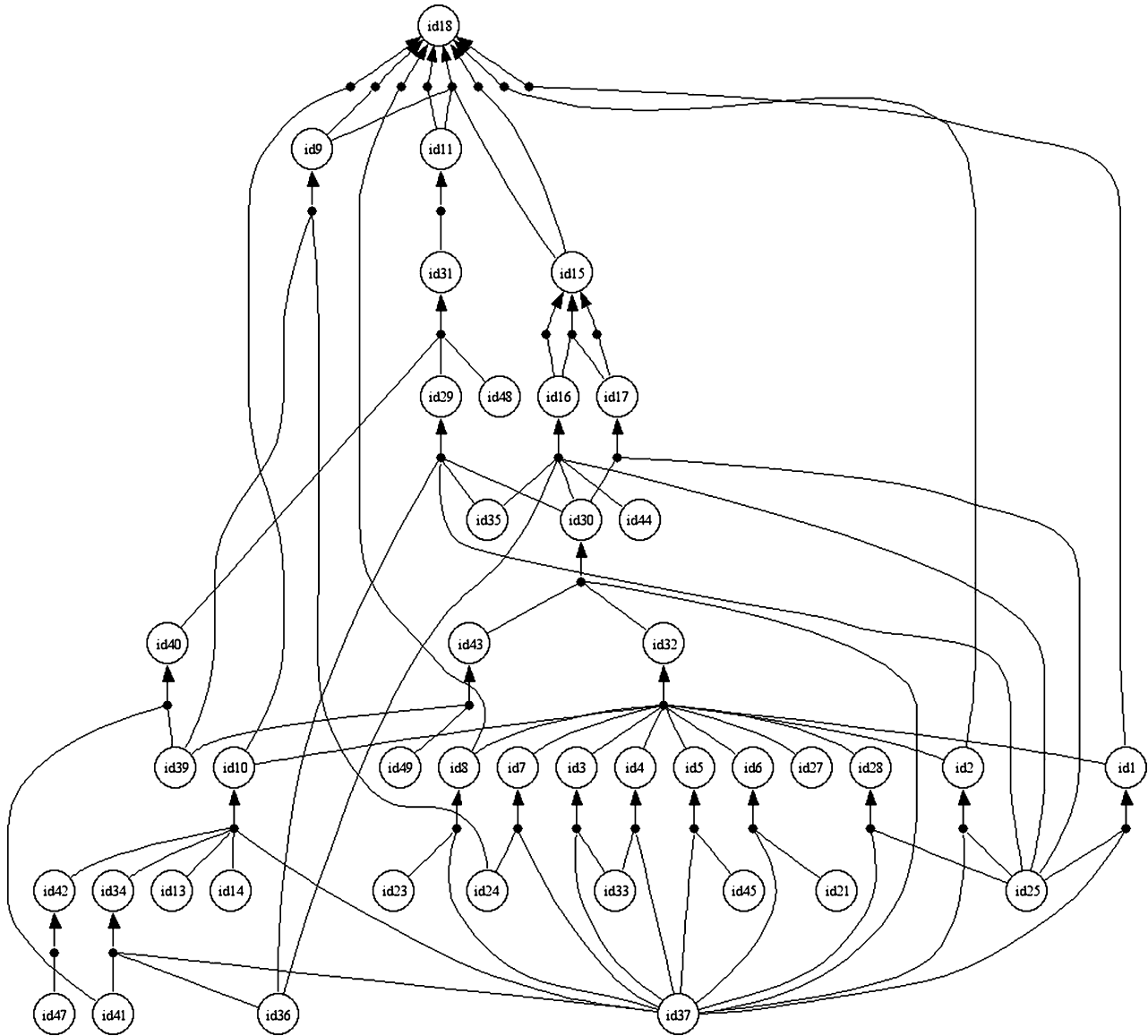


Fig. 8. The structure of information elements and their operations for the UWV case.

easily understood and that experiences fewer hand-off mistakes at run-time.

In our view, the case reveals that the results from seemingly reasonable criteria for defining activities as applied during a redesign project can be improved using the proposed metrics. At the same time, the application of the metrics does not interfere with the main objectives of the original redesign, as it keeps the processing order of operations intact and does not interfere with other objectives. So, this case illustrates that our heuristic provides added value on top of more high-level redesign objectives.

5. Tool support

Even though it provides concrete support for process designers, the calculation of the previously defined metrics for a given process design can be quite laborious. In our experimentation we found that human mistakes are easily

made when performing such calculations manually. Therefore, we developed a tool to support the design of a workflow process model based on the described coupling and cohesion metrics. We will refer to it as the *CoCoFlow* tool (*CO*hesion-*CO*upling metrics for work*FLOW* models). It is freely available for download and experimentation from <http://cocoflow.process-redesign.org>. The user interface of the *CoCoFlow* tool consists of three different sheets, i.e. the metrics sheet, the visualization sheet, and the XML-file sheet. Its main functionality is the calculation of the design metrics and the decision for the best design. This is supported by the metrics sheet, which is shown in Fig. 10.

The *CoCoFlow* tool reads an XML-file containing the information element structure and several process designs defined for this structure. The XML-file can be opened by using the 'Open XML-file button'. In the current version of the tool some example XML-files are available, such as the XML file for the IBG case described in Section 2. When the XML file is

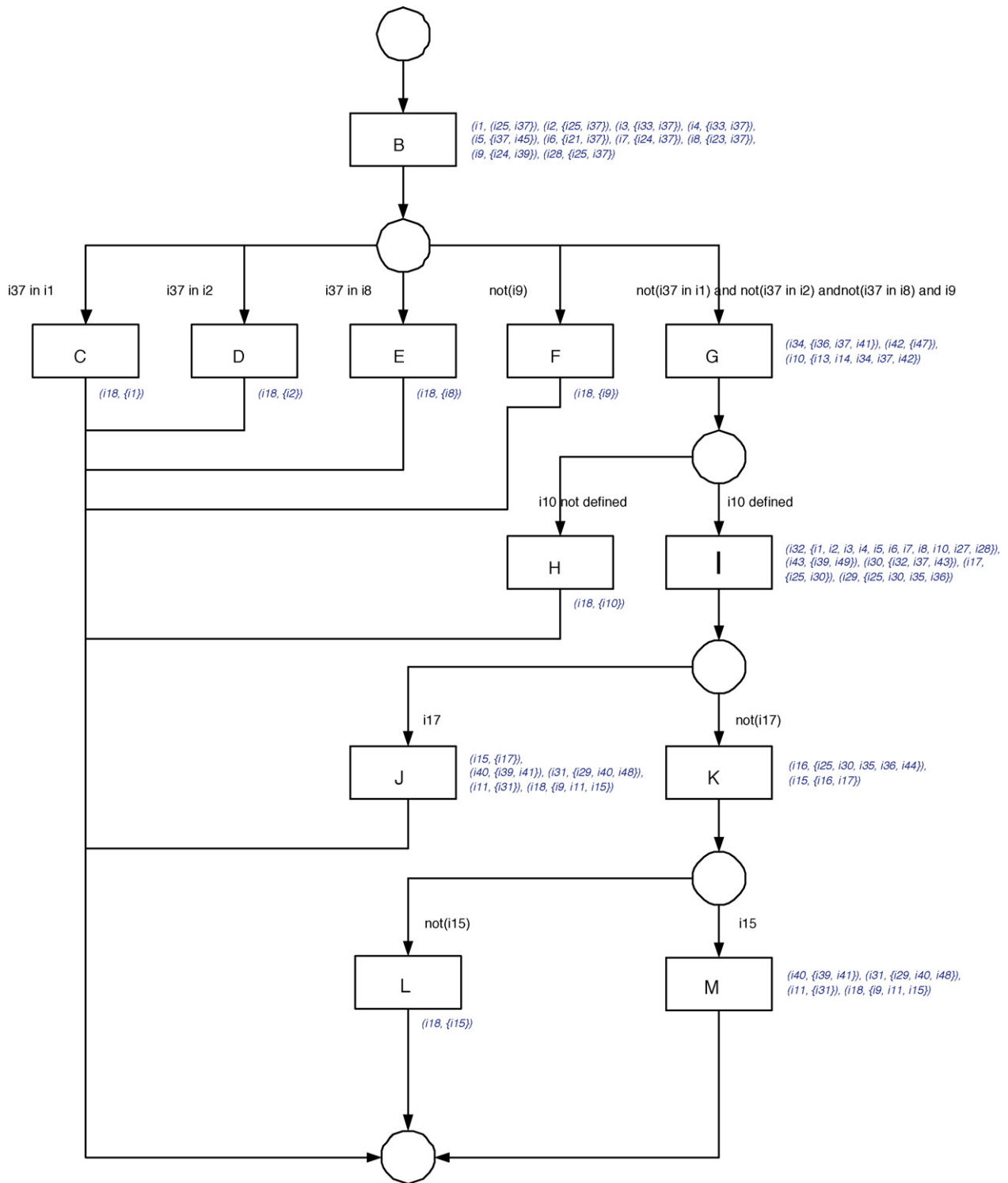


Fig. 9. The original design for the UWV case.

opened and read by the tool, the values for the metrics can be calculated. In the upper left corner the best design can be determined automatically by clicking the ‘find best design’ button. In the box below, the cohesion and coupling values for the selected design are shown and at the bottom left corner some checks on the selected design can be performed. At the right-hand side of the metrics sheet first some descriptions of the process and the selected design are given. In the lower right

corner the activity cohesion metrics can be determined for the selected design. In Fig. 10 all values for the first design of the IBG case are shown (cf. the outcomes of the calculations in Section 3.1).

Besides the calculation of the metrics, the *CoCoFlow* tool is able to generate a graphical depiction of the information element structure in the visualization sheet (see Fig. 11). By selecting a design and one of its activities, the tool

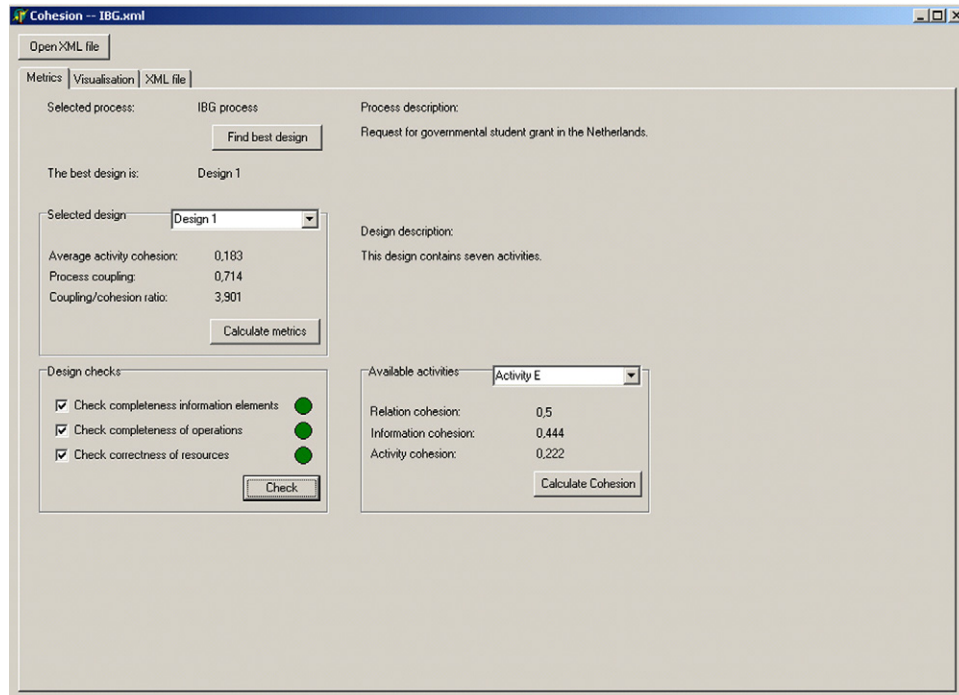


Fig. 10. Screenshot of the metrics sheet of the *CoCoFlow* tool.

highlights the corresponding information elements and operations. Finally, it is possible to edit and check the input XML file through the interface provided in the XML file sheet (Fig. 12).

6. Related work

The work reported in this paper relates to existing research in the area of workflow [5,19,22,32,33,58]. It is most related to

quality metrics in the domains of software engineering and workflow. Therefore, we discuss the work in these two domains in more detail.

6.1. Quality metrics in the software engineering domain

In the area of software engineering a wide variety of software quality metrics has been developed. The main purpose of software quality metrics is to obtain program designs that are

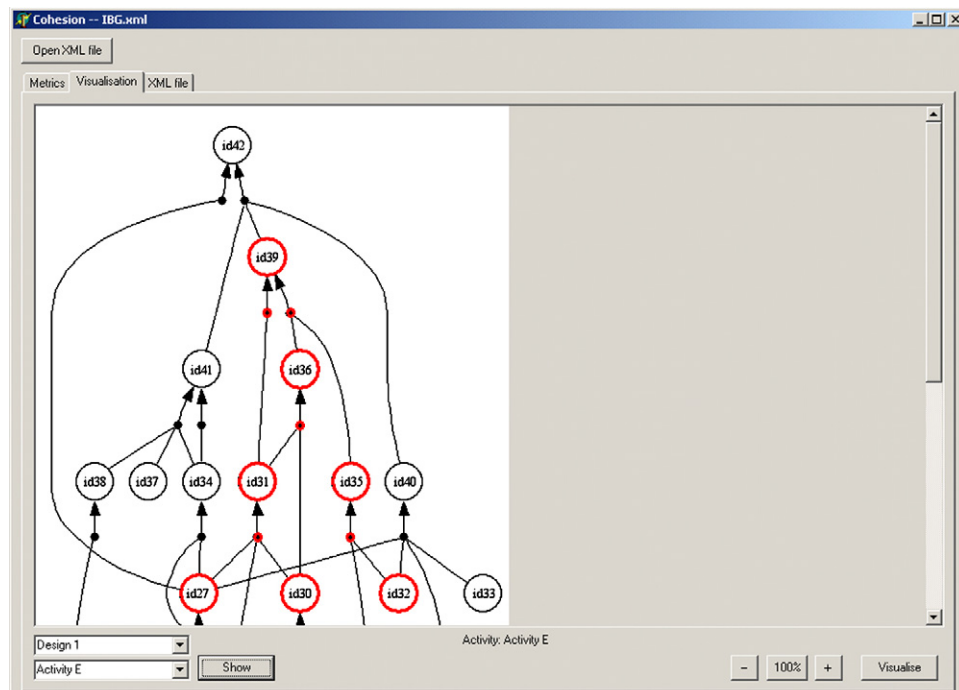


Fig. 11. Screenshot of the visualization sheet of the *CoCoFlow* tool.

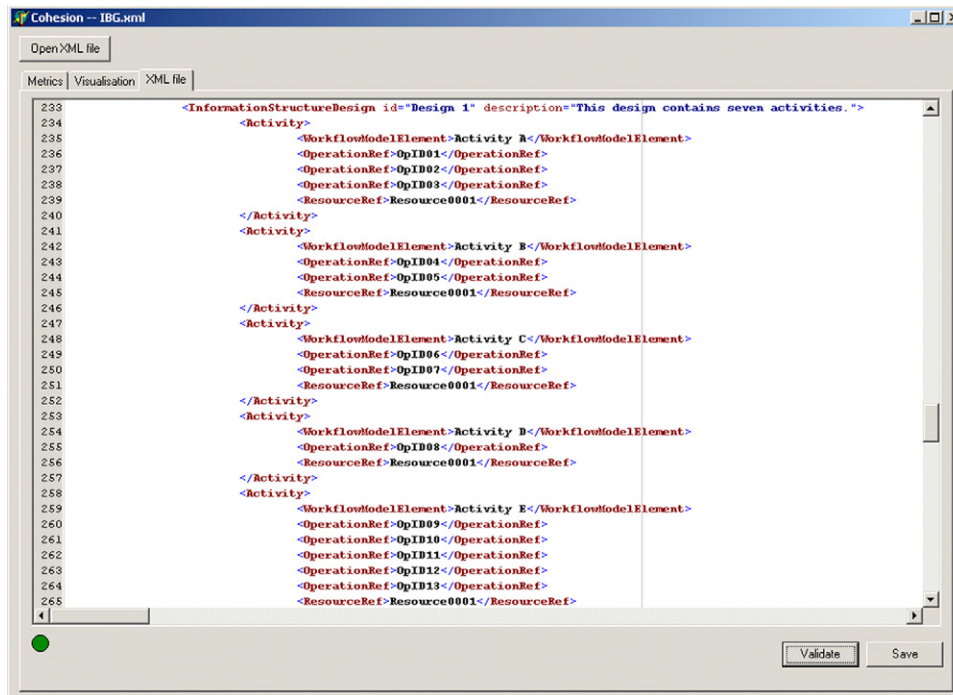


Fig. 12. Screenshot of the XML sheet of the *CoCoFlow* tool.

better structured. Some of the most important advantages of a structured design are, as pointed out in ref. [17], that (i) the overall program logic is easier to understand for both the programmers and the users and (ii) the identification of the modules is easier, since different functions are performed by different modules, which makes the maintenance of the software program easier.

According to refs. [17,51,54] the quality of a design is generally considered to be related to five design principles: coupling, cohesion, complexity, modularity, and size. Of these, coupling and cohesion are considered to be the most important ones:

- *Coupling* is measured by the number of interconnections among modules. Coupling is a measure for the strength of association established by the interconnections from one module of a design to another. It is hypothesized that programs with a high coupling will contain more errors than programs with lower coupling.
- *Cohesion* is a measure of the relationships of the elements within a module and is sometimes referred to as *module strength*. It is hypothesized that programs with low cohesion will contain more errors than programs with higher cohesion.

Researchers do not seem to agree on the relative importance of these two principles. In refs. [51,54], results of analyses are presented that indicate that coupling is the most influential of the design principles under consideration. However, in ref. [37] cohesion and coupling are considered as equally important. In this paper, we have adopted the latter view and combined notions of workflow process cohesion and coupling.

In addition, various researchers carried out studies to gather empirical evidence that coupling and cohesion quality metrics

do indeed improve the quality of a software design. Bieman and Kang, in particular, have shown various examples how cohesion metrics can be used to restructure a software design [11,28,29]. Also, in ref. [50] evidence is presented that low coupling and high strength (cohesion) are desirable. By calculating coupling/strength ratios of a number of routines in a software library tool it was found that routines with low coupling/strength ratios had significantly fewer errors than routines with high coupling/strength ratios. In ref. [12], a number of Fortran modules from a National Aeronautics and Space Administration project were examined. It was found that 50 percent of high-strength (high cohesion) modules were fault free, whereas only 18 percent of low-strength modules were fault free. No relationship was observed between fault rate and coupling. The results of ref. [12] indicate that modules with more descendants (modules that call many other modules) have a higher fault rate. Among the conclusions of this work there are two important results: (i) high cohesion reduces fault rate, and (ii) modules with many descendants (high coupling) are more fault prone than those with few. From these three empirical studies, we again conclude that both coupling and cohesion are important measures for the quality of a software design.

6.2. Quality metrics in the workflow domain

Because of the similarities between software programs and workflow processes, explained in Section 3 and Table 1, the application of similar quality metrics to the workflow field seems worthwhile to explore. We conducted a literature review on business process metrics and found out that, despite the vast literature on software engineering metrics, there is not much substantial research on business process metrics available yet. It is important to stress here that this paper's focus is on quality

metrics aimed at the *design* or *build time* aspect of workflow processes, while in the workflow domain “metrics” are often discussed as the evaluation of workflow data logs [34]. Various tools exist, e.g. PISA [36], PROM [40], and the ARIS Process Performance Manager [27], to determine values for *performance metrics*, such as average processing times, deadline reliabilities and risk indices, on the basis of historical execution data. We do not deal with such *run-time* metrics in this paper.

Research into quality metrics for the analysis of workflow designs is very limited, as also noted recently in ref. [14]. Yet, the issue of process granularity and its effect on the understandability and execution of process models was already explicitly identified in a seminal paper on process modelling [18]. The authors suggest that if humans should interpret a process model then it should be represented in larger units than in scripts for automatic execution, but “in many domains, descriptions for process scripts are presented to humans at too high a level of abstraction (large-grained) and they do not provide sufficient detail for guiding actual execution”. Clearly, this work identifies the design dilemma we addressed in this paper.

Interestingly, in refs. [7,8] the use of cohesion and coupling metrics was already suggested as a means to decompose a workflow model into sub-processes. However, no additional guidance is given. The suggestion does not seem to be motivated to increase the execution quality or understandability of the workflow model *per se*.

In ref. [9] an overview of guidelines is given that should be taken into account during the creation of a process model. Besides the basic guidelines of correctness, relevance, and economic efficiency, there is the additional guideline of *clarity*, which encompasses understandability. The authors state that “Without a readable, understandable, useful model all other efforts become obsolete”, stressing the relevance of this guideline. The proposed heuristic in this paper can be seen as support for following up on this guideline.

To our knowledge, the first concrete metric in the workflow domain is the *soundness* notion [1]. A workflow design is sound if each of its invocations (to deal with a specific case) can be guaranteed to terminate successfully. In particular, a sound workflow model does not contain deadlocks. Soundness can be determined on the basis of formal, structural characteristics of a workflow design. The Woflan software tool [56] is able to check soundness for workflow designs, allowing a variety of modelling techniques to be used. From a metrics point of view, soundness is a direct and boolean indicator for a specific *correctness* notion for workflow designs. It may also be seen as a minimum requirement that any workflow design should satisfy. This metric is not derived from any similar metric from software engineering, but is based on general notions used for the analysis of Petri nets, such as liveness and boundedness.

More recent work on the analysis of the workflow design quality has indeed been inspired by software practices. In ref. [24], the ISO/IEC 9126 Software Product Quality Model is used as a starting point for an evaluation framework for workflow designs. In this work, workflow process quality is a hierarchically layered structure, where metrics are used to measure the functionality, reliability, and usability of the

design. The various metrics are illustrated in ref. [24] with a realistic workflow case: For each metric a simple count takes place of the activities in the workflow design that satisfy it. This results in relative strengths and weaknesses of the design with respect to the various quality aspects. The authors argue that this approach can help organizations to select the most suitable alternatives to their processes. Although the breadth of the considered quality measures is considerable, exact definitions of the metrics are lacking.

Recent research is available on the adaptation of the five types of metrics from the software engineering domain to workflow processes. At this moment three surveys are available in which researchers describe the opportunities to apply software engineering metrics, and in particular complexity, to business process designs. They also investigate how these software engineering metrics can be adopted to the BPM field [15,23,31]. However, none of these metrics has been made very concrete and practical yet. Furthermore, they have not yet been tested and a number of limitations is identified for each metric.

In ref. [55] an overview is presented of the current state of the art in business process metrics. The same classification of the metrics over the five categories of coupling, cohesion, complexity, modularity and size was used as in the software engineering area. From this overview, we conclude that this field of research is just at its start and that there is a lot of potential for further development of business process metrics. Surprisingly, no work is done yet on the development of a modularity metric. Also, the classification and nomenclature is not yet very precise. For instance, Mendling uses a coupling metric as means to calculate complexity and Latva-Koivisto, Gruhn and Laue, and Cardoso et al. use size as a measure for complexity [15,23,31]. Perhaps, this classification of business process metrics should be revised when this area becomes more mature.

In our earlier work [43], which we briefly discussed in the introduction of this paper, we compared the application of a simple cohesion metric on workflow models to the decisions of 14 experienced workflow designers. We showed these designers’ ten design dilemmas inspired by practical workflow models, presenting them two alternative designs for each dilemma. For each dilemma the respondents indicated whether they preferred to combine the operations into one large activity or to split up the operations in two activities. Their answers were compared to the outcomes that the application of the simple cohesion metric suggested. A strong correspondence was established between these outcomes. In particular, the correlation between the average respondent score and the heuristics’ outcome turned out to be approximately 0.810. This significant result (assuming a two-tailed 99% confidence interval) supported the validity of the cohesion metrics and the viability of the underlying idea that we expanded on in this paper.

7. Conclusion

Our claim in this paper is that the proposed cohesion and coupling metrics will help designers in the creation of workflow

models that have a better execution quality and that are easier to understand by people. Understandability in its turn will cause a better maintainability of the workflow model. Our overall claim is based on the following assertions:

- The heuristic leads to *intuitive* outcomes. In our earlier experiments, i.e. the IBG toy example (Section 2), and an industrial case (Section 4), design alternatives are selected that incorporate activities which are neither too large nor too small. Such a balance is necessary because a great number of small activities will lead to a tightly coupled process with many error-prone hand-offs, while activities that are too large will lack cohesiveness and cannot be understood or performed well by humans. Involving a small group of experienced workflow designers in our research, we already showed that the cohesion metric conforms with expert decisions (see Section 6).
- Similar heuristics in software engineering are *effective*. Cohesion and coupling are good indicators for software quality with respect to the reduction of errors and maintainability. Considering workflow processes that are targeted at information processing and show similarities to software programs (see Section 3), similar results may be expected.
- The application of the heuristic in realistic workflow design settings is *feasible*. The computation of the cohesion and coupling metrics is supported by the *CoCoFlow* tool, which will limit the risk of human mistakes that can occur when manually determining these values (see Section 5). The focus on the information processing function is also compatible with the industry-strength workflow design method PBWD [2,42,45].

Further support for the viability of the metrics may be found in earlier suggestions by workflow researchers to use coupling and cohesion metrics for workflow model decomposition [7,8]. As we have argued in Section 6, we believe that our presented work delivers concrete design guidance, which is often absent in related work. Clearly, broader empirical evidence that support the effectiveness of the presented heuristic needs to be gathered. This will require extensive fieldwork, because the results of the workflow design with respect to its execution quality will only be measurable after the implementation of the design. In a typical workflow project, this period may span several months or even years. For the understandability of the workflow models, it is feasible to set up experiments which are not attached to actual projects. We are thinking about conducting experiments in the spirit of [48], where respondents will be asked to evaluate alternative designs under laboratory conditions.

We have used software engineering metrics as a source of inspiration, but it should be noted that software engineers use metrics for a much wider range of properties than we have considered here. In this paper, we have aimed for a modest goal with respect to workflow process design, focusing only on execution quality and maintainability issues. These properties have led to the most tangible results in the software engineering

domain. A comprehensive quality concept for a workflow design should perhaps encompass additional aspects such as reliability, security, interoperability, etc. [24].

One of the drawbacks of the presented heuristic is its elaborate mechanics. In contrast to the metrics as proposed in ref. [13], it is not likely that the metrics will acquire popularity on the basis of their simplicity (as, for instance, the “number of lines code” metric in software engineering). The development of the *CoCoFlow* tool can be expected to counter this effect for some part. Fortunately, in settings where the PBWD method is applied, the step towards the application of the proposed heuristic is rather small because the fine-grained analysis of information elements and operations has already taken place. PBWD has been adopted as one of the standard innovation methods by the management consultancy firm Deloitte and is frequently applied.

From a scientific perspective, the greatest challenge lies in the fact that the heuristic does not suggest any clustering or ordering of information elements itself. An extension of the heuristic so that it can efficiently generate (semi-) optimal activity definitions itself would generate a quantum leap in this domain.

Finally, we would like to note that this approach is not limited to the application domain of Petri nets. This method can be used together with any process modelling language (such as EPC, IDEF, or Data Flow Diagrams) as long as the product structure is properly described. However, in the cases which do not have a suitable product description or in which the product cannot be described as a network, other quality metrics in the workflow domain (Section 6.2) may still be very useful for design guidance.

Acknowledgements

This research is supported by the Technology Foundation STW, Applied Science Division of NWO and the technology programme of the Dutch Ministry of Economic Affairs.

Appendix A. Request for student grant example

Table A.1

Description of information elements of the information element structure in Fig. 1

Number	Description
1	Last name applicant
2	First name(s) applicant
3	Address applicant
4	Telephone number(s) applicant
5	Sex applicant
6	Social Security Number applicant
7	Country where applicant lives
8	Birth place applicant
9	Account number applicant
10	Identification number applicant (student grant institute)
11	Signature applicant
12	First day of study
13	Date from which applicant wants to receive a scholarship
14	Signature father applicant
15	Signature mother applicant

Table A.1 (Continued)

Number	Description
16	Identification number father of applicant
17	Identification number mother of applicant
18	Period or year for which applicant requests a grant
19	Date of request
20	Birth date applicant
21	Nationality applicant
22	The kind of student grant the applicant requests (There are four possibilities for its value: (1) a basic scholarship (2) basic scholarship and supplementary scholarship (3) basic scholarship, supplementary scholarship and loan (4) basic scholarship and loan)
23	Age applicant
24	Social Security Number father of applicant
25	Reference year for tax authorities
26	Social Security Number mother of applicant
27	Applicant has a right to receive a student grant
28	Income father of applicant
29	Income mother of applicant
30	Income applicant's parents
31	Applicant has a right to receive a supplementary scholarship
32	Kind of health insurance of applicant
33	Living situation applicant
34	Applicant has a right to receive a loan from the student grant institute
35	Maximum amount that can be received for supplementary scholarship
36	Parental contribution
37	The amount of loan the applicant requests
38	Maximum amount of loan
39	The amount of supplementary scholarship that is assigned to applicant
40	The amount of basic scholarship assigned to applicant
41	The amount of loan that is assigned to applicant
42	The total amount of student grant assigned to applicant

References

- [1] W.M.P. van der Aalst, The application of Petri nets to workflow management, *The Journal of Circuits, Systems and Computers* 8 (1) (1998) 21–66.
- [2] W.M.P. van der Aalst, On the automatic generation of workflow processes based on product structures, *Computers in Industry* 39 (2) (1999) 97–111.
- [3] W.M.P. van der Aalst, Reengineering knock-out processes, *Decision Support Systems* 30 (4) (2001) 451–468.
- [4] W.M.P. van der Aalst, P.J.S. Berens, Beyond workflow management: product-driven case handling, in: S. Ellis, T. Rodden, I. Zigurs (Eds.), *International ACM SIGGROUP Conference on Supporting Group Work (GROUP 2001)*, ACM Press, New York, 2001, pp. 42–51.
- [5] W.M.P. van der Aalst, K.M. van Hee, *Workflow Management: Models, Methods, and Systems*, MIT Press, Cambridge, MA, 2002.
- [6] W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, A.P. Barros, Workflow patterns, *Distributed and Parallel Databases* 14 (1) (2003) 5–51.
- [7] L. Baresi, F. Casati, S. Castano, M. Fugini, I. Mirbel, B. Pernici, WIDE workflow development methodology, in: *Proceedings of International Joint Conference on Work Activities Coordination and Collaboration*, 1999, pp. 19–28.
- [8] L. Baresi, F. Casati, S. Castano, M. Fugini, P. Grefen, I. Mirbel, B. Pernici, G. Pozzi, Workflow design methodology, in: P. Grefen, B. Pernici, G. Sanchez (Eds.), *Database Support for Workflow Management: the WIDE Project*, Kluwer Academic Publishers, 1999, pp. 47–94.
- [9] J. Becker, M. Rosemann, C. von Uthmann, Guidelines of business process modeling, in: W.M.P. van der Aalst, J. Desel, A. Oberweis (Eds.), *Business Process Management: Models, Techniques, and Empirical Studies*, Lecture Notes in Computer Science 1806, 2000, pp. 30–49.
- [11] J.M. Bieman, B.-K. Kang, Measuring design-level cohesion, *IEEE Transactions on Software Engineering* 24 (2) (1998) 111–124.
- [12] D.N. Card, V.E. Church, W.W. Agresti, An empirical study of software design practices, *IEEE Transactions on Software Engineering* 12 (2) (1986) 264–271.
- [13] J. Cardoso, How to measure the control-flow complexity of web processes and workflows, in: L. Fischer (Ed.), *Workflow Handbook*, Lighthouse Point, 2005, pp. 199–212.
- [14] J. Cardoso, Control-flow complexity measurement of processes and Weyuker's properties, in: *Proceedings of the Sixth International Enformatika Conference (IEC 2005)*, 2005, pp. 213–218.
- [15] J. Cardoso, J. Mendling, G. Neumann, H.A. Reijers, A discourse on complexity of process models, in: J. Eder, S. Dustdar, et al. (Eds.), *BPM 2006 Workshops*, Lecture Notes in Computer Science 4103, Springer-Verlag, Berlin, 2006, pp. 115–126.
- [17] S.D. Conte, H.E. Dunsmore, V.Y. Shen, *Software Engineering Metrics and Models*, Benjamin/Cummings Publishing Company, Inc., 1986.
- [18] B. Curtis, M.I. Kellner, J. Over, Process modelling, *Communications of the ACM* 35 (9) (1992) 75–90.
- [19] M. Dumas, W.M.P. van der Aalst, A.H.M. ter Hofstede, *Process-Aware Information Systems: Bridging People and Software Through Process Technology*, Wiley, 2005.
- [22] L. Fischer, *Workflow Handbook*, Future Strategies, 2005.
- [23] V. Gruhn, R. Laue, Complexity metrics for business process models, in: W. Abramowicz, H.C. Mayr (Eds.), *9th International Conference on Business Information Systems (BIS 2006)*, volume 85 of *Lecture Notes in Informatics*, 2006, 1–12.
- [24] A.S. Guceglioglu, O. Demiros, Using software quality characteristics to measure business process quality. *Proceedings of the 3rd International Conference on Business Process Management (BPM 2005)*, Lecture Notes in Computer Science 3649, 2005, pp. 374–379.
- [25] M. Hammer, *Reengineering work: don't automate, obliterate*, Harvard Business Review (1990) 70–91.
- [26] M. Hammer, J. Champy, *Reengineering the Corporation: A Manifesto for Business Revolution*, Harper Business, 1993.
- [27] IDS Scheer. ARIS process performance manager: measure, analyze and optimize your business process performance. Product Description and Fact Sheet, 2002 (<http://www.ids-scheer.com>).
- [28] B.-K. Kang, J.M. Bieman, Using design cohesion to visualize, quantify, and restructure software, in: *8th International Conference on Software Engineering and Knowledge Engineering*, Knowledge Systems Institute, Skokie, IL, 1996, pp. 222–229.
- [29] B.-K. Kang, J.M. Bieman, A quantitative framework for software restructuring, *Journal of Software Maintenance* 11 (1999) 245–284.
- [31] A.M. Latva-Koivosto, Finding a Complexity Measure for Business Process Models, Helsinki University of Technology, Systems Analysis Laboratory, 2001.
- [32] F. Leymann, D. Roller, *Production Workflow: Concepts and Techniques*, Prentice Hall, 2000.
- [33] D.C. Marinescu, *Internet-Based Workflow Management: Towards a Semantic Web*, John Wiley & Sons, 2002.
- [34] M. McLellan, Workflow Metrics—One of the Great Benefits of Workflow Management, in: H. Österle, P. Vogler (Eds.), *Praxis des Workflow-Management*, Braunschweig, 1996, pp. 301–318.
- [36] M. zur Muehlen, M. Rosemann, Workflow-based process monitoring and controlling: technical and organizational issues, in: *Proceedings of the 33rd Hawaii International Conference on System Science (HICSS-33)*, 2000, pp. 1–10.
- [37] G.J. Myers, *Composite/Structured Design*, Van Nostrand Reinhold, New York, NY, 1978.
- [38] A. Orlicky, Structuring the bill of materials for MRP, *Production and Inventory Management* (December 1972) 19–42.
- [39] E.A.H. Platier, A logistical view on business processes: concepts for business process redesign and workflow management, PhD Thesis, Eindhoven University of Technology, Eindhoven, 1996.

- [40] Process Mining Research. Website <http://www.processmining.org>, accessed on February 21, 2006.
- [41] H.A. Reijers, Product-based design of business processes applied within the financial services, *Journal of Research and Practice in Information Technology* 34 (2) (2002) 34–46.
- [42] H.A. Reijers, Design and control of workflow processes: business process management for the service industry, *Lecture Notes in Computer Science* 2617, Springer-Verlag, Berlin, 2003.
- [43] H.A. Reijers, A cohesion metric for the definition of activities in a workflow process, in: *Proceedings of the Eighth CAiSE/IFIP8.1 International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design 2003*, 2003, pp. 116–125.
- [44] H.A. Reijers, S. Limam Mansar, Best practices in business process redesign: an overview and qualitative evaluation of successful redesign heuristics, *Omega: The International Journal of Management Science* 33 (4) (2005) 283–306.
- [45] H.A. Reijers, S. Limam Mansar, W.M.P. van der Aalst, Product-Based Workflow Design, *Journal of Management Information Systems* 20 (1) (2003) 229–262.
- [46] H.A. Reijers, I.T.P. Vanderfeesten, Cohesion and coupling metrics for workflow process design, *Proceedings of the 2nd International Conference on Business Process Management (BPM 2004)*, *Lecture Notes in Computer Science* 3080, 2004, pp. 290–305.
- [47] K. Sarshar, P. Loos, Comparing the control-flow of EPC and Petri net from the end-user perspective, *Proceedings of the 3rd International Conference on Business Process Management (BPM 2005)*, *Lecture Notes in Computer Science* 3649, 2005, pp. 434–439.
- [48] A. Seidmann, A. Sundararajan, The effects of task and information asymmetry on business process redesign, *International Journal of Production Economics* 50 (2–3) (1997) 117–128.
- [49] R.W. Selby, V.R. Basili, Analyzing error-prone system structure, *IEEE Transactions on Software Engineering* 17 (2) (1991) 141–152.
- [50] M. Shepperd, *Software Engineering Metrics*, Vol. I, Metrics and Validations, McGraw-Hill, 1993.
- [51] W.P. Stevens, G.J. Myers, L.L. Constantine, Structured design, *IBM Systems Journal* 13 (2) (1974) 115–139.
- [52] D.A. Troy, S.H. Zweben, Measuring the quality of structured designs, *Journal of Systems and Software* 2 (1981) 113–120.
- [53] I. Vanderfeesten, J. Cardoso, J. Mendling, H.A. Reijers, W.M.P. van der Aalst, Quality metrics for business process models, in: L. Fischer (Ed.), *BPM and Workflow Handbook*, Future Strategies Inc., Lighthouse Point, FL, USA, 2007, pp. 179–190.
- [54] H.M.W. Verbeek, T. Basten, W.M.P. van der Aalst, Diagnosing workflow processes using Woflan, *The Computer Journal* 44 (4) (2001) 246–279.
- [55] The Workflow Management Coalition. Website <http://www.wfmc.org>, accessed on February 21, 2006.
- [56] M. Xenos, D. Stavrinoudis, K. Zikouli, D. Christodoulakis, Object-oriented metrics—a survey, in: *Proceedings of the Federation of European Software Measurement Association (FESMA) 2000*, 2000, pp. 1–10.



Irene Vanderfeesten is a PhD candidate in the Information Systems group of the Department of Technology Management at Eindhoven University of Technology, The Netherlands. She received her MSc in computer science from the same university (2004) and is now working in the research project ‘Intelligent software tools for workflow process design’. Her research interests include workflow management, business process redesign, product based workflow design, and human aspects of information systems. She published papers at several international conferences and has received the BPTrends Best Student Paper Award 2004.



Hajo A. Reijers is an assistant professor of business process management at Eindhoven University of Technology. He received his PhD and MSc in computer science from the same institute and has worked for several management consultancy firms, most recently as a manager within Deloitte. His research interests include business process redesign, workflow management systems, and discrete event simulation. He published articles in *Information Systems*, *Journal of Management Information Systems*, *Computer Supported Cooperative Work*, *Omega: The International Journal of Management Science*, *Computers in Industry*, and other journals.



Wil van der Aalst is a full professor of information systems at the Technische Universiteit Eindhoven (TU/e) having a position in both the Department of Mathematics and Computer Science and the Department of Technology Management. Currently he is also an adjunct professor at Queensland University of Technology (QUT) working within the BPM group there. His research interests include workflow management, process mining, Petri nets, business process management, process modeling, and process analysis. Wil van der Aalst has published more than 80 journal papers, 12 books (as author or editor), 200 refereed conference publications, and 25 book chapters. Many of his papers are highly cited and his ideas have influenced researchers, software developers, and standardization committees working on process support. He has been a co-chair of many conferences including the International Conference on Cooperative Information Systems, the International Conference on the Application and Theory of Petri Nets, and the Business Process Management Conference, and is an editor/member of the editorial board of several journals, including the *Business Process Management Journal*, the *International Journal of Business Process Integration and Management*, the *International Journal on Enterprise Modelling and Information Systems Architectures*, *Computers in Industry*, *Lecture Notes in Business Information Processing*, and *Transactions on Petri Nets and Other Models of Concurrency*.