“Precise is better than Light” A Document Analysis Study about Quality of Business Process Models

Gianna Reggio, Maurizio Leotta, Filippo Ricca

Abstract:

Business process modelling is often used in the initial phases of traditional software development to reduce faulty requirements and as starting point for building SOA based applications.

Often, modellers produce business process models without following recognized guidelines and opt for “light” models where nodes representing the actions are simply decorated with natural language text. The potential consequence of this practice is that the quality of built business process models may be low.

In this paper, we propose a method based on manual transformations to detect flaws in “light” business process models expressed as activity diagrams. Using that method we have executed a document analysis study with 14 business process models taken by books and websites. Preliminary results of this study show that almost all the analysed business process models contain errors and style violations (precisely 92% of them).

Digital Object Identifier (DOI):

`http://dx.doi.org/10.1109/EmpiRE.2011.6046257`

"Precise is better than Light"
A Document Analysis Study about Quality of Business Process Models

Gianna Reggio, Maurizio Leotta and Filippo Ricca
Dipartimento di Informatica e Scienze dell'Informazione - DISI
Università di Genova
16146 Genova, Italy
{ *gianna.reggio* | *maurizio.leotta* | *filippo.ricca* }*@disi.unige.it*

***Abstract*—Business process modelling is often used in the initial phases of traditional software development to reduce faulty requirements and as starting point for building SOA based applications.**

Often, modellers produce business process models without following recognized guidelines and opt for "light" models where nodes representing the actions are simply decorated with natural language text. The potential consequence of this practice is that the quality of built business process models may be low.

In this paper, we propose a method based on manual transformations to detect flaws in "light" business process models expressed as activity diagrams. Using that method we have executed a document analysis study with 14 business process models taken by books and websites. Preliminary results of this study show that almost all the analysed business process models contain errors and style violations (precisely 92% of them).

***Keywords*-Document Analysis, Business Process Models, Quality, Activity Diagrams, Flaw Taxonomy**

I. Introduction

It is well-known that the main motivation to face conceptual modelling is to reduce the chances of developing faulty requirements. Recently, an empirical study has shown that business processes have become the central objects in many conceptual modelling efforts [1], e.g., to support their documentation. Moreover, often, business process models constitute the starting point for building SOA based applications. For example, [2] proposes a method that defines how a business process should be transformed into services and how these services should collaborate to fulfil business goals.

Recently, in literature, a number of notations have been proposed to model business processes, e.g., BPMN (Business Process Modeling Notation) [3], Petri Nets [4] and UML (Unified Modeling Language) [5], [6]. UML represents a natural choice for modelling business processes since it provides the activity diagrams. Activity diagrams model the flow of activities, making them ideal to model business processes.

In favour of the UML notation is its flexibility allowing the modeller to choose the preferred degree of precision/abstractiveness to build business process models. Different options are available ranging from "light" styles, where nodes and arcs of the activity diagrams are simply decorated by natural language text, to more rigorous solutions, where nodes and arcs are complemented by expressions in a formal language. "Light" activity diagrams are simple to sketch but their inherent ambiguity complicates comprehension [7] and thus the communication among participants. Conversely, a more precise/rigorous style is more complex to follow, but the ambiguity should be reduced. Moreover, "precise" activity diagrams can be more easily transformed into executable models (e.g., expressed in BPEL [8]), and allow to fully automate such transformations.

In this paper, we propose: (1) a method based on (a form of) inspection and manual transformations to detect errors and style violations in "light" business process models and (2) a document analysis study[1] [9], where we have applied the proposed method to several business processes taken by authoritative books and Websites for evaluating its effectiveness.

Preliminary results of our study show that almost all the analysed business process models contain errors and style violations.

The remainder of the paper is organized as follows: Section II presents, by means of a running example, the adopted method to transform "light" models in to "precise" ones. Section III introduces the Research Questions and illustrates the followed procedure, while Section IV shows and discusses the achieved results. Section V presents relevant related literature concerning quality of business process models. Final remarks conclude the paper.

II. The Adopted Method

We have devised a method with the aim to get a UML "precise" business process model starting from the corresponding "light" version.

In a nutshell, the method requires to identify the participants and the business objects of the business process model

[1] Document analysis is the systematic examination of documents such as books, websites, paintings and laws. The focus of the analysis should be a critical examination, rather than a mere description, of the documents.

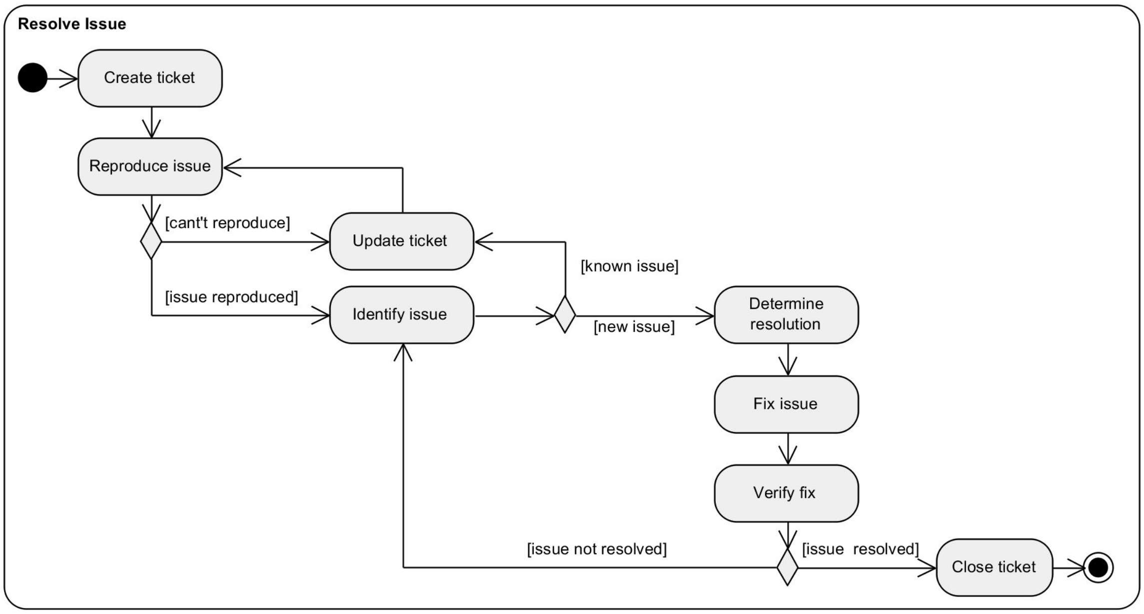

Figure 1. Resolve Issue case: original Activity Diagram

and to model them by means of a UML class diagram. Then, each node and arc of the target "light" activity diagram, decorated by natural language text, is respectively transformed in calls of operations and OCL expressions.

Following the "precise style" introduced in [10], we illustrate how "to make precise" a UML activity diagram with an example. In the meantime, we will show how this activity leads us to discover possibly errors and style violations present in the source diagram.

The case is "Resolve Issue"; we have found it on a Website presenting UML, where it is shortly described in this way: "*After ticket is created by some authority and the issue is reproduced, issue is identified, resolution is determined, issue is fixed and verified, and ticket is closed, if issue was resolved*". The original activity diagram is shown in Figure 1.

First of all, the method requires to use well-formed UML diagrams; in this case there are no problems and we cannot find any syntactic error.

Then, the method requires to determine the participants of the business process (either business workers, i.e., human beings, or hardware/software systems) and the business objects, and to model them by means of a UML class diagram. This diagram for the "Resolve Issue" case is shown in Figure 2, where proper stereotypes define the classes corresponding to the different kinds of entities involved in the business process (e.g., ≪businessWorker≫ and ≪businessObject≫). The source diagram does not offer any clue about the participants, while it is clear that there is the business object "Ticket", the accompanying note suggests that the ticket is created by some authority, and we assume that a developer will take care of handling the ticket. The participants and the objects of the business process are listed in a note added to the activity diagram (see the bottom of Figure 3).

The basic tasks of the process, modelled in the source diagram by action nodes, e.g., Create ticket, will be modelled by calls of operations of participant classes stereotyped by ≪T≫ (for Task), such as "TICKET = AUTH.createsTicket()" (createsTicket returns the new ticket). Operations may return also data values, as for example reproduces that returns a boolean (see Figure 2).

During the analysis of the source diagram, we inferred that a ticket has associated an issue and a resolution (Figure 1), and we added this knowledge to the class diagram with new classes (Resolution and Issue) and associations. Moreover we added some new other ≪T≫ operations to the class diagram (e.g., fixes that takes a resolution).

The ≪T≫ operations can then be defined more precisely by means of pre-post conditions. For example we require that "updates" changes the issue of the ticket, or that initially the ticket has no associated resolution.

During the definition of the post condition for updates, we detected that two different activities in the source diagram were called with the same name. Look indeed to Update ticket: when the issue cannot be reproduced the issue part of the ticket must be modified, but when the issue is already know the ticket must be modified by merging it with the corresponding old ones. Thus, in the precise diagram we model it with two different operations corresponding to these two different activities. In the source diagram there is another flaw. The designer has tried to reuse the balloon labelled Update ticket, thus in the case of issue

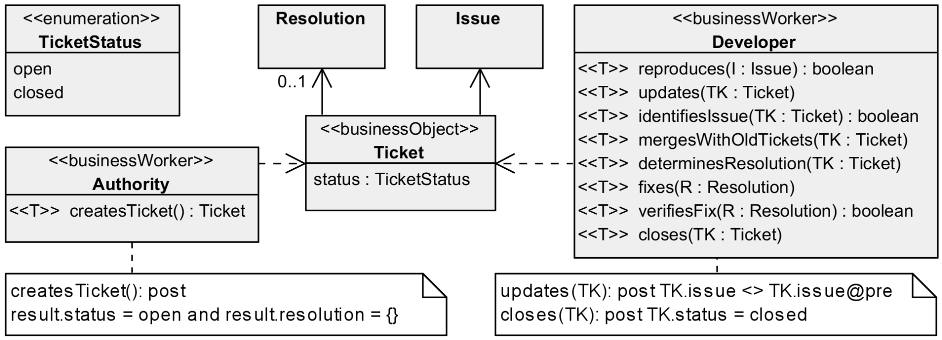

Figure 2. Resolve Issue case: precise model (Class Diagram)

already known there is a non terminating cycle.

The precise style requires also that for each decision node there should be the corresponding merge node (except when a branch reaches a termination node). In the source diagram of Figure 1 there are three decision nodes without explicit merge nodes, so we have added them (see Figure 3). This modification lead us to discover more flaws in the source diagram, indeed the activity node labelled by Reproduce issue has two input edges, and so it waits for two control tokens resulting in a deadlock. This semantic error (we call it *absence of merge node*) may be due to an old feature of the UML (UML 1.x) which assumed an implicit merge node [11], and it is present other two times in the source diagram (Update ticket and Identify issue resulting in other two possibilities for deadlock).

Finally, the precise style imposes that the flow of the time is depicted vertically, at least the "main/good/correct" flow, while the exceptional/erroneous cases are depicted horizontally. Applying this style convention to the diagram, we obtain a longer picture but truly more readable. Indeed the new layout allowed us to detect another flaw: in the source diagram when a resolution does not work it is required to modify the issue instead of the "bad" resolution.

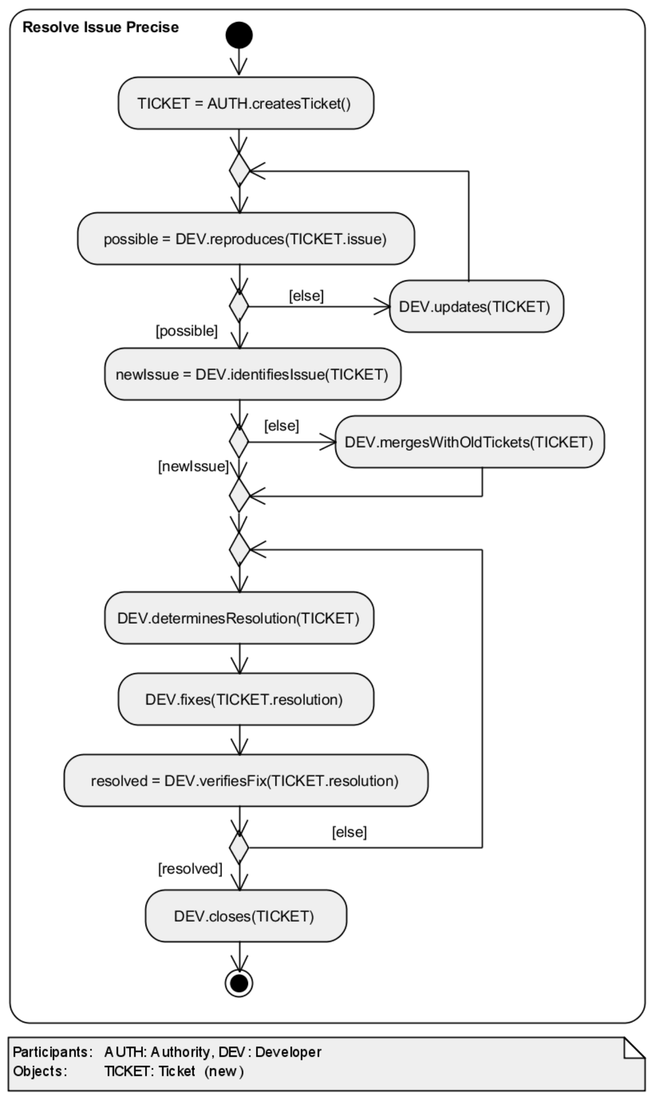

Figure 3. Resolve Issue case: precise model (Activity Diagram)

III. Document Analysis Study

A document analysis study has been conducted to evaluate effectiveness and application effort of the method presented in the previous section.

The main aim of our study is answering the following research questions:

> **RQ1**: *What is the effectiveness of our method in revealing errors and style violations in Business Process Models represented with activity diagrams?*
>
> **RQ2**: *What is the effort required to apply our method?*

The first question (**testing effectiveness**) deals with the ability of our approach to reveal errors and style violations. By answering **RQ1**, we can understand the strength of our method and deduce, at least in our sample, the percentage of erroneous vs. correct "light" business process models. This,

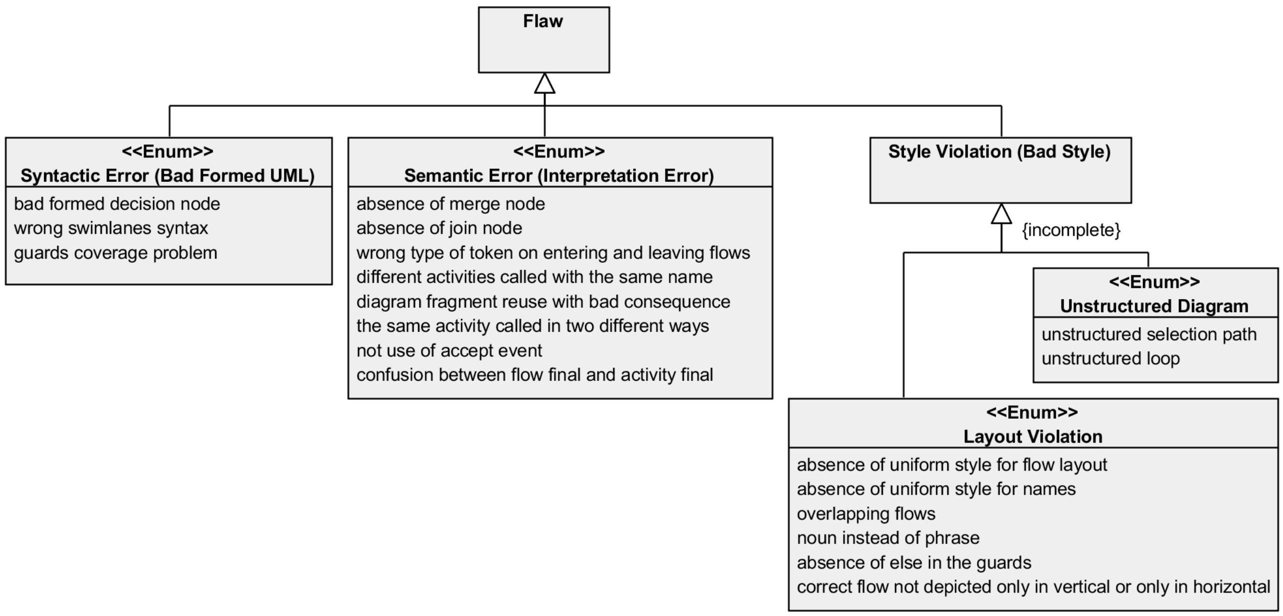

Figure 4. Populated Flaw Taxonomy

can give us an indirect measure of how much is easy/difficult making errors and violations in “light” business process models.

Question **RQ2** (**testing effort**) deals with the effort required by the application of our approach. For instance, even if a given approach can reveal more errors/violations, it may be too onerous and thus not applicable in practice.

A. Procedure

The document analysis study has been performed as follows:

1) Consulting the literature, we defined an initial high level taxonomy of possible errors and style violations for business process models expressed by means of UML activity diagrams;
2) We collected 14 “light” business process models (our dataset);
3) For each selected “light” business process model, we applied our method obtaining a “precise” model corresponding to the “light” version. Each found error or violation was annotated in a summarizing table and the initial taxonomy was populated.

B. Taxonomy of Errors and Violations

We based the first level of our taxonomy on the Lindland’s quality framework [12]. Lindland’s framework relates different aspects of modelling to three concepts: syntax, semantics and pragmatics. These concepts are described as follows (citing from [13]). All models should be *syntactically* correct, thereby adhering to the rules of the modelling language (in our work, in some cases UML 1.x while in others UML 2.0). All models should represent their intended *semantic* meanings and should do so consistently. All models should have good *aesthetics*, demonstrating the creativity and farsightedness of their modellers. This means that software models should be symmetric, complete and pleasing in what they represent. Consequently, our taxonomy first level (see Figure 4) is composed of three classes: syntactic error, semantic error and style violation.

C. Data Collection

We randomly selected our diagrams: (1) consulting several authoritative books in Software engineering (e.g., Software Engineering. Theory and Practice - 2nd edition - S.L. Pfleeger) and in business process modelling, and (2) surfing the Web using the Google search engine.

Simplifying, we classified the models in three types: toy example (if used for explaining a concept or illustrating a feature of the modelling language, e.g., swimlanes or object nodes in activity diagrams), realistic (if used as a case study in a scientific paper, technical report or book) and real (if corresponding to a real business process).

Finally, the objects of our study were 14 business process models: 11 toy examples, 1 realistic model and 2 real models.

D. Application of the Method

For each selected business model, we applied the following procedure. First, the model was copied in Visual Paradigm[2]. Second, the method presented in Section 2 was

[2] a UML modeller covering all kinds of UML diagram types. See http://www.visual-paradigm.com/

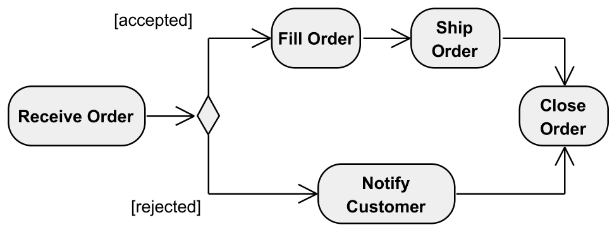

Figure 5. Absence of Merge Node

applied by one of the authors (G. Reggio). Finally, the obtained results, i.e., the "precise" version and the list of flaws (if any), was checked by another author (M. Leotta) to avoid as much as possible human errors and subjective classifications. To reach a consensus among the authors several meetings were held. The effort has been measured by the first author as time to copy/duplicate the original "light" model in Visual Paradigm and as time to complete the transformation (i.e., as time to produce the "precise" version). It was recorded directly by the author noting down start and stop time.

The final result of this manual activity was threefold:

- for each business model in the dataset, we obtained a "precise" model corresponding to the "light" version taken in input;
- a taxonomy populated by the detected errors and violations;
- a summarizing table containing the detected errors and violations (see next section).

IV. RESULTS

A. Preliminary Flaw Taxonomy

Figure 4 shows the empirically populated flaw taxonomy. For space reasons we cannot describe all the detected flaws. Some flaws are simple to understand (e.g., *overlapping flows*), while others are less simple. For example, one that deserves an explanation is the semantic error *absence of merge node*, already discussed in Section II. Figure 5 shows the error. The absent merge node bringing two flows together to close an order in Figure 5 is required, because without it, both flows would need to arrive before closing the order, which would never happen [11].

The *Unstructured diagram* category points out only the difference between unstructured selection paths and unstructured loops. It represents a simplification of the five cases reported in [14]. It is important to note that style violations of type *Unstructured diagram* are more serious that those of type *Layout* because they make difficult the translation of business process models into executable models (e.g., expressed in a language as BPEL) that offer structured-programming constructs only. All the descriptions of the flaws included in the taxonomy, plus some examples to understand them, are available starting from http://softeng.disi.unige.it/2011-PreciseVSLight.php.

B. Detected Errors and Violations

Table I details some information about the 14 analysed activity diagrams. The Table reports: the source type (Book, WebSite and Scientific Paper), the kind (Toy, Realistic and Real as explained previously in this paper), the number of nodes, swimlanes and object nodes. Moreover, the Table shows also: the version of UML used originally to describe the activity diagram (where the version was not explicitly indicated we have deduced it from the year of publication of the source[3]) and the amount of time (in minutes) spent to copy it faithfully with respect to the original and the amount of time spent to transform it. In the case of the `Single Sign-On` we have not produced the precise model (and thus the time is not present in Table I). The source diagram is about a specific protocol, and thus to be able to make it precise we should study the protocol itself quite well, which requires a huge amount of time. Furthermore, using an activity diagram to model a protocol is not the best choice (a sequence diagram could have been a better choice). It is possible to download all the models (before and after the method application) from the following URL: http://softeng.disi.unige.it/2011-PreciseVSLight.php.

Table II reports the reference for each activity diagram used in our study while Table III reports the detected errors and style violations per diagram. The Table is structured following the flaw taxonomy described in the previous section.

C. Discussion

The results reported in Table III show the effectiveness of our method in revealing errors and violations (**RQ1**). We found at least an error or violation in 13 out of 14 selected activity diagrams (i.e., 92%) and at least a serious semantic error in 12 out of 14 activity diagrams (i.e., 85%).

The most frequent flaw, present with at least one occurrence in 10 out of 14 activity diagrams (i.e., 71%), is absence of merge node; it is also, by far, the more frequent semantic error. We hypothesize that it can be a consequence of a misunderstanding of the concept/use of tokens in the activity diagrams. The second flaw for number of occurrences (9 occurrences) is *absence of else in the guard*; this is generally a minor problem (really it is a style violation not an error). However, if the guards in a decision node are not carefully set, from this kind of minor problem could arise as consequence another more serious, the *guard coverage problem*, that may cause a deadlock. This is the case that we found in the `Meet a Client` model where a decision node with two guards "appointment onsite" and "appointment offsite" follows the action "Call Client and

[3]in two cases we were not able to deduce the UML version

	Inventory Class	Document Creation	Meet a Client	Resolve Issue	Single Sign-On	Business Proposal	Withdraw Money	Software Activation	Enroll in University	Passenger Checks In	Order Mgmt	Cancer Register	Account Creation	Getting a Plane
Source Type	Book	Book	Book	WebSite	WebSite	WebSite	WebSite	WebSite	WebSite	WebSite	WebSite	Paper	Book	Book
Kind	Toy	Toy	Toy	Toy	Toy	Toy	Toy	Real	Toy	Toy	Toy	Real	Toy	Realistic
# Nodes	10	13	11	13	19	18	14	19	11	13	11	49	11	27
# Swimlanes	0	0	0	0	5	0	0	3	0	2	3	4	0	4
# Object Nodes	0	0	0	0	0	0	0	6	0	0	0	8	0	3
UML Version	1.x	2.0	2.0	2.3	2.3	1.x	1.x	2.3	2.0	-	-	2.0	2.0	2.0
Duplic. time	7	11	8	9	20	12	22	15	10	13	10	42	6	23
Transf. time	10	21	23	38	-	61	23	32	24	18	17	109	12	57

Table I
MODELS DESCRIPTION (DUPLICATION AND TRANSFORMATION TIMES ARE EXPRESSED IN MINUTES)

set up Appointment”. This decision node violates non only *absence of else in the guard* but also cause a *guard coverage problem*; in fact, if the client does not accept to set up any appointment the process gets stuck.

By answering RQ1, we can deduce the percentage of erroneous vs. correct “light” business process models. This, give us an indirect measure of how much is easy/difficult making errors and violations in “light” business process models. Given that the number of errors and style violations in the analysed set of models is large, we can speculate that producing a correct “light” activity diagram representing a business process model is difficult. This could be due to a series of factors. For example, the lack of a formal semantics of UML, the several different versions of UML (from UML 1.1 to UML 2.3) and the high number of available kinds of node and alternative notations (e.g., pins instead of object nodes).

The effort required to create a precise version of the source activity diagrams seems not to be very high (**RQ2**). In general, we have observed that the time required to transform the “light” version in the corresponding precise one is between 2 and 3 times the time required to copy/duplicate it (at least, this is true with an expert modeller as G. Reggio). Indeed, from the data in Table I we found a regression line $y = ax + b$ with $a = 2.36$ and $b = 0.01$ fitting discretely the data ($R^2 = 71\%$[4]).

The main implication of these results for practice is a simple advice to modelling consultant: to pay attention to “light” activity diagrams. At a glance, they seem very simple and easy to understand but often they contain subtle flaws that could bring to different interpretations and meanings. The Flaw taxonomy sketched in Figure 4 can be used to avoid or at least reduce the more common flaws.

[4] R^2 measures how well a regression approximates the real data points; R^2=100% indicates that the regression perfectly fits the data.

D. Threats to Validity

The main limitations to this investigation are: (i) the selection bias regarding the considered books/websites and then the selected business process models, and (ii) the possibly inaccuracy in flaw classification and method application. However, the flaw classification and the obtained “precise” models was checked, rechecked and discussed among all the authors.

Another issue is generalization. Given the small dataset and the nature of this kind of investigation, we can not generalize the obtained findings to all the population of business process models and for all the skill levels of modellers. Nevertheless, we believe that these results are interesting and constitute a good starting point for further investigations.

V. RELATED WORKS

In literature, there are several empirical works concerning the quality of Business Process Models and more in general of UML models. However, to the best of our knowledge, this is the first work that tries to propose: (1) a method to detect errors and style violations in “light” business process models expressed as activity diagrams and (2) an empirically found flaw taxonomy for business process models.

The paper [15] provides a set of recommendations on how to build a process model from scratch and for improving existing process models. Each of the guidelines is built on empirical research described in other papers. We agree with the seven guidelines proposed in [15] but we observed that in our dataset they are rarely followed (e.g, “G6” use verb-object activity labels and “G4” model as structured as possible).

The book [13] goes in the same direction of [15] but extends the topic to UML 2.0 diagrams. In particular, it lists strengths, weaknesses, objectives and traps of several kind of UML diagrams and provides a set of syntax, semantic and aesthetic checks. Concerning the activity diagrams,

	Information
Inventory Class	S. L. Pfleeger. “Software Engineering: Theory and Practice”, 2nd Edition., Prentice-Hall, 2001 - Figure 6.25
Document Creation	J. Schmuller. “Sams Teach Yourself UML in 24 Hours”, Complete Starter Kit, 3rd Edition, Sams, 2004 - Figure 11.5
Meet a Client	J. Schmuller. “Sams Teach Yourself UML in 24 Hours”, Complete Starter Kit, 3rd Edition, Sams, 2004 - Figure 11.6
Resolve Issue	www.uml-diagrams.org/activity-diagrams-examples.html
Single Sign-On	www.uml-diagrams.org/activity-diagrams-examples.html
Business Proposal	www.ibm.com/developerworks/rational/library/2802.html
Withdraw Money	www.ibm.com/developerworks/rational/library/2802.html
Software Activation	www.uml-diagrams.org/activity-diagrams-examples.html
Enroll in University	www.agilemodeling.com/style/activityDiagram.htm
Passenger Checks In	http://sourcemaking.com/uml/modeling-business-systems/external-view/activity-diagrams
Order Mgmt	www.analisi-disegno.com/businessmodeling/Business ModelingUML.pdf
Cancer Register	Shiki Naomi, Ohno Yuko, Fujii Ayumi, Murata Taizo, Matsumura Yasushi. “Unified Modeling Language (UML) for hospital-based cancer registration processes”. Asian Pacific journal of cancer prevention: APJCP 2008;9(4):789-96.
Account Creation	R. Miles, K. Hamilton. “Learning UML 2.0 - A Pragmatic Introduction to UML”, O’Reilly Media, 2006 - Figure 3-2
Getting a Plane	M. J. Chonoles, J. A. Schardt. “UML 2 For Dummies”, John Wiley & Sons Inc, 2003 - Figure 13-5

Table II
REFERENCES

the more interesting are the semantic checks. During our transformational work on models (from “light” to “precise”) we implicitly applied some of the checks proposed in [13], for example: “*Check for semantically correct dependencies between activities and check to ascertain the correctness of the names of the activities and their corresponding meaning within the business domain*”.

A number of other papers address the quality issue of UML models evaluating the impact of best practices, such as the use of modelling conventions. Modelling conventions are similar to coding conventions, but they apply to the model instead of the code. For example, the effectiveness of modelling convention and their impact is empirically investigated in [16]. The results of that study indicate that modelling conventions decrease the defect density of the model but are not able to improve clarity, completeness, and validity of the information. The authors interpret this result concluding that mere properties of the model (e.g., syntax, design and layout) are not sufficient to improve its quality.

		Inventory Class	**Document Creation**	**Meet a Client**	**Resolve Issue**	**Single Sign-On**	**Business Proposal**	**Withdraw Money**	**Software Activation**	**Enroll in University**	**Passenger Checks In**	**Order Mgmt**	**Cancer Register**	**Account Creation**	**Getting a Plane**
Syntactic	bad formed decision node	X	-	X	-	-	-	-	-	-	-	X	-	-	-
	wrong swimlanes syntax	-	-	-	-	-	-	-	-	-	X	X	-	-	-
	guards coverage problem	-	-	X	-	-	-	-	-	-	-	-	-	-	-
	SubTotal	1	0	2	0	0	0	0	0	0	1	2	0	0	0
Semantic	absence of merge node	X	X	X	X	-	X	X	-	X	-	X	X	-	X
	absence of join node	X	-	-	-	-	X	-	-	-	-	-	-	-	X
	wrong type of token on entering and leaving flows	-	-	-	-	-	-	-	X	-	-	-	-	-	-
	different activities called with the same name	-	-	-	X	X	-	-	-	-	-	-	-	-	-
	diagram fragment reuse with bad consequence	-	-	-	X	-	-	X	-	-	-	-	-	-	-
	the same activity called in two different ways	-	X	-	-	-	-	-	-	-	-	-	-	-	-
	not use of accept event	X	-	-	-	-	X	-	-	-	-	-	-	-	-
	confusion between flow final and activity final	-	-	-	-	-	-	-	-	-	-	-	X	-	X
	SubTotal	3	2	1	3	1	3	2	1	1	0	1	2	0	3
Layout	absence of uniform style for flow description	-	-	-	X	-	-	-	-	X	-	X	-	-	-
	absence of uniform style for names	-	X	X	-	-	-	-	-	-	-	-	-	-	-
	overlapping flows	-	X	-	-	-	-	-	-	-	-	-	-	-	-
	noun instead of phrase	-	-	-	-	X	X	-	-	-	-	-	-	-	-
	absence of else in the guards	X	X	X	X	X	X	-	-	-	-	X	X	-	X
	correct flow not depicted only in vertical or only in horizontal	-	-	-	X	X	-	-	-	-	-	-	-	-	-
	SubTotal	1	3	2	3	3	2	0	0	1	0	2	1	0	1
Unstruct.	unstructured selection path	-	-	-	-	-	-	X	-	-	-	X	X	-	X
	unstructured loop	-	-	-	-	-	X	-	-	X	-	-	X	-	-
	SubTotal	0	0	0	0	0	1	1	0	1	0	1	2	0	1
Total		5	5	5	6	4	6	3	1	3	1	6	5	0	5

Table III
DETECTED ERRORS AND VIOLATIONS

Even if UML is the de facto standard for modelling software systems its lack of a formal semantics and its complexity cause the risk of a lot of practical defects. Lange and Chaudron in [17] conducted a study trying to quantify the distribution of defect types in real industrial models. The study shows that the number of defects found in industrial UML models is very large. This result is consistent with our findings.

VI. Conclusion

In this paper, we have conducted a document analysis study to estimate effectiveness and application effort of our method for detecting flaws in business process models expressed as activity diagrams.

Preliminary results show the effectiveness of our manual method in revealing errors and style violations. Overall, we found 55 flaws (whereof 23 semantic errors) in the 14 analysed models. As far as effort is concerned, we can say that the effort for applying the method is not too much expensive (2 or 3 times the time to copy/duplicate the corresponding "light version"), al least this is true when an expert modeller applies it.

An interesting additional result, to confirm with further empirical studies, is that producing correct "light" activity diagrams representing business process models seems to be a difficult task. This is direct consequence of the fact that the number of errors and style violations in the analysed set of authoritative selected models is so large.

Future work will be devoted to refine our method and extend the current dataset, adding more real and realistic models to it. We believe that quality in modelling is of paramount importance; for this reason we intend continue to investigate in this direction. We would like also propose a similar method for UML state machine diagrams and conduct a similar study with them.

References

[1] I. Davies, P. Green, M. Rosemann, M. Indulska, and S. Gallo, "How do practitioners use conceptual modeling in practice?" *Data & Knowledge Engineering*, vol. 58, pp. 358–380, September 2006.

[2] M. Rychly and P. Weiss, "Modeling of service oriented architecture: From business process to service realisation," in *International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE)*, 2008, pp. 140–146.

[3] Object Management Group (OMG), "Business Process Model and Notation (BPMN) - Version 2.0," *OMG Final Adopted Specification*, 2011.

[4] E. Sivaraman and M. Kamath, "On the use of Petri nets for business process modeling," in *11th Annual Industrial Engineering Research Conference*, 2002.

[5] E. D. Nitto, L. Lavazza, M. Schiavoni, E. Tracanella, and M. Trombetta, "Deriving executable process descriptions from UML," in *Proceedings of the 24th International Conference on Software Engineering*, ser. ICSE '02. New York, NY, USA: ACM, 2002, pp. 155–165.

[6] E. Astesiano, G. Reggio, and F. Ricca, "Modeling business within a UML-based rigorous software development approach," in *Concurrency, Graphs and Models*, ser. Lecture Notes in Computer Science, P. Degano, R. De Nicola, and J. Meseguer, Eds. Springer Berlin / Heidelberg, 2008, vol. 5065, pp. 261–277.

[7] F. Di Cerbo, G. Dodero, G. Reggio, F. Ricca, and G. Scanniello, "Precise vs. ultra-light activity diagrams - an experimental assessment in the context of business process modelling," in *International Conference on Product Focused Software (PROFES)*, 2011, 291–305.

[8] Organization for the Advancement of Structured Information Standards (OASIS), "Web Services Business Process Execution Language - Version 2.0," *OASIS Standard*, 2007.

[9] L. F. Prior, "Document analysis : The sage encyclopedia of qualitative research methods," *SAGE Publications*, 2008.

[10] G. Reggio, F. Ricca, E. Astesiano, and M. Leotta, "On business process modelling with the UML: a discipline and three styles," Dipartimento di Informatica e Scienze dell'Informazione (DISI) - University of Genova, Italy, Tech. Rep. DISI-TR-11-03, April 2011. [Online]. Available: http://softeng.disi.unige.it/tech-rep/TECDOC.pdf

[11] C. Bock, "UML 2 activity and action models part 3: Control nodes," *Journal of Object Technology*, vol. 2, no. 6, pp. 7–23, November 2003.

[12] O. I. Lindland, G. Sindre, and A. Sølvberg, "Understanding quality in conceptual modeling," *IEEE Software*, vol. 11, pp. 42–49, March 1994.

[13] B. Unhelkar, *Verification and Validation for Quality of UML 2.0 Models*. Wiley-Interscience, 2005.

[14] M. H. Williams and H. L. Ossher, "Conversion of unstructured flow diagrams to structured form," *The Computer Journal*, vol. 21, no. 2, pp. 161–167, 1978.

[15] J. Mendling, H. A. Reijers, and W. M. P. van der Aalst, "Seven process modeling guidelines (7PMG)," *Information and Software Technology*, vol. 52, pp. 127–136, February 2010.

[16] C. Lange, B. DuBois, M. Chaudron, and S. Demeyer, "An experimental investigation of UML modeling conventions," in *Model Driven Engineering Languages and Systems*, ser. Lecture Notes in Computer Science, O. Nierstrasz, J. Whittle, D. Harel, and G. Reggio, Eds., vol. 4199. Springer Berlin / Heidelberg, 2006, pp. 27–41.

[17] C. Lange and M. R. Chaudron, "Defects in industrial UML models — a multiple case study," in *Workshop on Quality in Modeling at MODELS 2007*, September 2007.