Business Process Modelling: Five Styles and a Method to Choose the Most Suitable One

Gianna Reggio, Maurizio Leotta, Filippo Ricca, Egidio Astesiano

Abstract:

A software developer facing a modelling task may follow different styles at different levels of abstraction and precision, to better cope with the aims and the potential users of the model. We address the problem of modelling the business processes by means of UML activity diagrams, and present five styles differing in the precision level, from the Ultra-Light style, where the nodes and the edges of the activity diagram are decorated by freely-formed text, to precise styles where instead OCL and UML actions are used. Then, we propose a practical empirical method for choosing the most suitable style depending on the context in which the models will be used (why, when, where, how long, by whom).

Digital Object Identifier (DOI):

http://dx.doi.org/10.1145/2424563.2424574

Copyright:

© ACM, 2012. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in Proceedings of 2nd International Workshop on Experiences and Empirical Studies in Software Modeling (EESSMod 2012).

http://dx.doi.org/10.1145/2424563.2424574

Business Process Modelling: Five Styles and a Method to Choose the Most Suitable One

Gianna Reggio, Maurizio Leotta, Filippo Ricca, Egidio Astesiano
Dipartimento interscuola di Informatica, Bioingegneria, Robotica e Ingegneria dei Sistemi (DIBRIS)
Università di Genova, Italy
{ gianna.reggio | maurizio.leotta | filippo.ricca | egidio.astesiano }@unige.it

ABSTRACT

A software developer facing a modelling task may follow different styles at different levels of abstraction and precision, to better cope with the aims and the potential users of the model. We address the problem of modelling the business processes by means of UML activity diagrams, and present five styles differing in the precision level, from the Ultra-Light style, where the nodes and the edges of the activity diagram are decorated by freely-formed text, to precise styles where instead OCL and UML actions are used. Then, we propose a practical empirical method for choosing the most suitable style depending on the context in which the models will be used (why, when, where, how long, by whom).

Categories and Subject Descriptors:
D.2.2 [**Software Engineering**]: Design Tools and Techniques

General Terms: Design, Documentation.

Keywords: UML, Business Process Modelling, Styles.

1. INTRODUCTION

The Unified Modeling Language (UML) is used to model many disparate aspects of different software and systems, in all the phases of the development process, with different aims, and by different kinds of stakeholders (e.g., business analysts and developers). This is possible because the UML offers a large number of constructs and allows to leave out any detail. Thus, a modeller when facing with a modelling task may follow different styles[1]. This has been recognized since long time. For example, Fowler proposes three different ways to build UML models: *UML as Sketch*, *UML as Blueprint*, and *UML as Programming Language* [4]. Within the UML, the activity diagrams are used to model and visualize the flows of control (and of data) in different entities, such as systems, objects, use cases, and operations.

[1]A particular manner or technique by which something is done, created, or performed (Merriam Webster's Dictionary)

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
EESSMOD '12, October 01-05, 2012, Innsbruck, Austria
Copyright 2012 ACM 978-1-4503-1811-2/12/10 ...$15.00.

Activity diagrams have been also used to model business processes [3, 5]. Modellers may follow different styles when modelling the business processes using the activity diagrams, e.g., the few guidelines suggested in [1]. An examination of the publicly available activity diagrams shows that the large majority of the modellers are completely undisciplined and produce activity diagrams without following any style. Moreover, except [1], no other relevant proposals are available. Also prompted by the participation in various research projects conducted in cooperation with the industry, we have then defined various styles for modelling the business processes with the UML activity diagrams that differ for the degree of precision of the produced models. These styles range from the Ultra-Light, where no guidelines drive the modellers, to styles for producing precise[2] models, where OCL and UML actions are exploited to decorate arcs and nodes, respectively. Each style is motivated by some specific modelling activities made in some specific context by specific persons.

Then, we have tried to evaluate these five styles. For example, for what concerns the comprehension [7] we got some empirical evidence that a precise model is easier to comprehend. By another empirical investigation we have found that producing models following the Ultra-Light style may results in making many mistakes and errors, that can be detected and corrected when revising such models following a precise style [6]. The precise models seem to be "better"; however, they have also problematic aspects, e.g., their production requires more effort and work of people with a good UML knowledge. Thus, in this paper we address the problem of finding the most suitable style when modelling the business processes using the UML. We propose a method, based on our experience in the context of business process modelling, that, on the basis of the context in which the modelling will be done, evaluates the five styles by giving a value to their suitability in such context.

The main contributions of this paper are:
– a detailed presentation of the five styles (described in this paper exhaustively for the first time) ranging from the lighter to the more precise by means of a running example;
– a method able to help the decision makers to choose the most suitable style depending on the context in which the models will be used.

We present the five styles for the business modelling with the UML in Sect. 2. In Sect. 3 we propose our method to evaluate the five styles for choosing the most suitable one. Finally, in Sect. 4 we draw some conclusions.

[2]Exactly or sharply defined or stated (Merriam Webster's Dictionary)

2. THE FIVE STYLES

In this paper we assume the common intuitive meaning of *business process*, i.e.: "*A progression of tasks (activities, interactions, ...) that involve two or more entities, and create or add value to the organization's activities. In a sequential process, each step is dependent on occurrence of the previous step; in a parallel process, two or more steps can occur concurrently*"; and we will use the following terminology:
– the *(business process) participants* are the active entities performing the various tasks. We distinguish the participants that are human beings (and thus capable of autonomous activities) from those corresponding to software and hardware systems.
– the *(business process) objects* are the entities over which the activities of the process are performed, obviously these entities are passive, i.e., they are not able to do any activity by themselves.
– the *(business process) data* are the data used in the various tasks.

To present the five styles for Business Process modelling (shortly *BP modelling*), we will use as a running example the business process corresponding to ordering in an e-commerce site (EC), briefly described as follows:

A client sends an order. If the client is not already registered, (s)he will be asked to register to the site, if (s)he refuses the order will be cancelled. Then the order will be sent to the warehouse, which will prepare the package, and in the meantime either the handler of the credit cards or Paypal will be contacted (depending on the preferences of the client, expressed at the registration time) to get the payment; after the package will be sent; finally the carrier will inform that the package has been delivered, and the order will be archived.

In this section, we introduce five styles for BP modelling using the UML: Ultra-Light, Light, Disciplined, Precise Conceptual and Precise Operational, where the degree of precision is minimum in the first style and maximum in the last two. Fig. 1 presents the structure of the models of the business processes produced following the five styles.

2.1 The Ultra-Light Style

The simplest models are those produced following the *Ultra-Light style*: they just consist of an activity diagram produced without following any guideline, using[3] action nodes, control nodes, edges, and time and accept events, where the nodes and the edges are decorated by natural language text fragments freely formed; we can better name it *No-Style*, since the modeller is completely free to produce the activity diagram as s(he) likes. The Ultra-Light style is the most commonly used, and obviously it is the easiest and fastest to follow, but the Ultra-Light models are also easily full of mistakes, see, e.g., [6], and cannot be used for any kind of post elaboration.

[3]Swimlanes and object nodes are not treated here for space reasons.

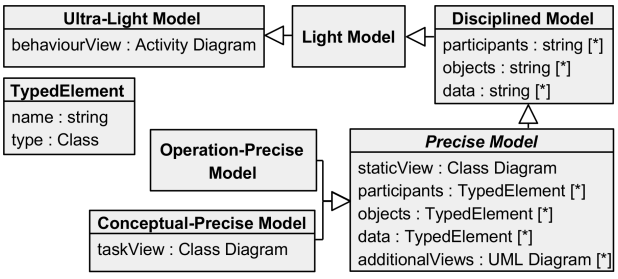


Figure 1: BP model structure for the five styles

2.2 The Light Style

The *Light style* only imposes few restrictions (listed in the following) on the use of the visual constructs of the activity diagrams and on their layout, whereas the decorations of nodes and edges are still completely unconstrained: they are just natural language text fragments.
– For each decision node there must be a matching merge node and similarly for any fork node there must be a matching join node (exception can be made whenever a flow leaving the decision/fork node ends with a final node).
– One outgoing edge from a decision node must be labelled with the "else" guard.
– The flowing of the tokens should be depicted vertically, and the edges leaving a decision node should be depicted as follows: the edge corresponding to the regular/correct course of the events should be vertical, whereas the alternative corresponding to an error or an exceptional case should be depicted horizontally.

Notice that it may happen that the sentences defining the activities may be either in active or passive form (e.g., "Clerk fills the form" and "Form is filled"), and that the entity executing the activity may be precisely determined or left undefined (e.g., "Form becomes filled"); in other cases it is possible that nominal sentences are used instead of verbal phrases ("Filling the form"). Also the objects over which the business process activities are performed may be described in different ways, for example by a substantive (e.g., "Form", "The form") or by a qualificative sentence (e.g., "Client form", "Filled form").

Fig. 2 presents the Light model of the EC business process. As required, it is an activity diagram satisfying the con-

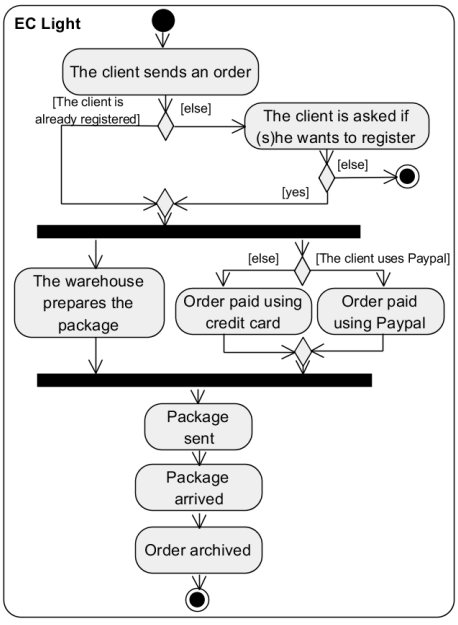


Figure 2: EC: Light Model

straints just introduced[4]. The various tasks are denoted by natural language sentences having different structure; some are active and makes explicit the subject (e.g., The client sends an order), some others are in the passive form (Order archived) and so no information is given on who will perform the task. Notice how the relationship between the task The client is asked if (s)he wants to register and the subsequent decision node with an edge with guard yes is completely based on the reader understanding of the meaning of an English sentence.

2.3 The Disciplined Style

A business process model that follows the *Disciplined style* consists of (see Fig. 1): the participant/object/data lists (written using CAPITAL LETTERS), and an activity diagram describing the process behaviour, where:
– the action nodes are decorated by tasks described by simple natural language sentences having the form: "*subjects + present tense verb + object complements + other complements*" or "*subjects + present tense passive form + other complements*", where the *subjects* and the *object complements* are either participants or business objects of the process, and the data may appear in *other complements*.
– the guards on the edges leaving the decision nodes must be qualificative sentences about some of the participants/ objects/data of the business process, e.g., "*X is ...*", "*X has ...*", "$X \geq 10$".

Notice that the participants/objects are roles for the entities taking part in the business process and not specific individuals.

To determine the tasks of a business process it is important to keep in mind that they are assumed to be atomic in the context of the model of such process (i.e., it is not important/relevant to detail them further in term of actions of the various participants), but it is not always true that they correspond to elementary actions of the participants. For example it is ok to have a task of the form "The SELLER and the CLIENT exchanges the CONTRACT", as well as to give a more detailed model where the tasks are instead "The SELLER sends the CONTRACT" and "The CLIENT receives the CONTRACT".

The passive form must be used whenever who will execute the task is either not relevant or not known. "CLIENT pays INVOICE" is an example of active sentence, whereas "ORDER is archived" is a case of passive sentence; both follow the above constraints. The use of the passive style should be quite careful. If we do not want to describe or we do not know who are the participants of the business process, we can represent several tasks using the passive form. The resulting business process model will be quite abstract, and it may be then refined by transforming the passive sentences into active ones, after having determined the subjects.

Note that the guards cannot be actions of someone, for example "X answered yes" or "X accepts" cannot be a guard; in this case there should be a task corresponding to give an answer and then the decision will be about the answer, thus the guard will have the form "answer = Yes".

Fig. 3 shows the Disciplined model of the EC business process. Notice that in this model all the sentences are in active form, e.g., EC archives ORDER and EC registers CLIENT, because the style requests to find the participants lead us

[4]For space reasons the two edges leaving the second decision node are horizontal instead of almost vertical.

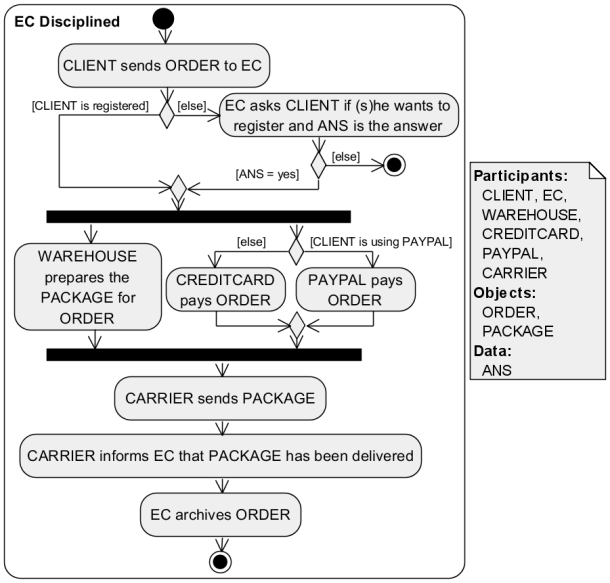


Figure 3: EC: Disciplined Model

to detect the presence of EC, the system supporting the e-commerce. To represent the answer of the client to the registration proposal we have used a process data ANS; in this way the relationships between the guards and the previous task is clear, whereas in the Light model it was completely left to the reader's intuition.

2.4 The Precise Style with Conceptual Tasks

The *Precise style with Conceptual Tasks style* (shortly *Precise Conceptual style)* for BP modelling requires to describe the participants, the objects and the data precisely by means of a class diagram, named *static view*, and to use an activity diagram to model the behaviour of the process, whereas the conditions on the edges leaving the decision nodes will be OCL expressions, and the action nodes will be decorated by elements of special classes stereotyped by ≪task≫, introduced by a class diagram named *task view*. Thus, a Precise Conceptual model consists of (see Fig. 1): *i)* a static view, i.e., a class diagram introducing the classes needed to type its participants, objects and data, *ii)* the participant/object/data lists, *iii)* a task view, i.e., a class diagram introducing the task classes, *iv)* and an activity diagram representing its behaviour, satisfying the following constraints.

a) The classes in the static view must be stereotyped by ≪object≫ (business process objects), ≪businessWorker≫ and ≪system≫ (business process participants distinguished in autonomous entities, human beings and hardware/software systems); datatypes may be also included in this class diagram. The elements of those classes may be described using the many tools offered by the UML, for example constraints and behavioural diagrams.

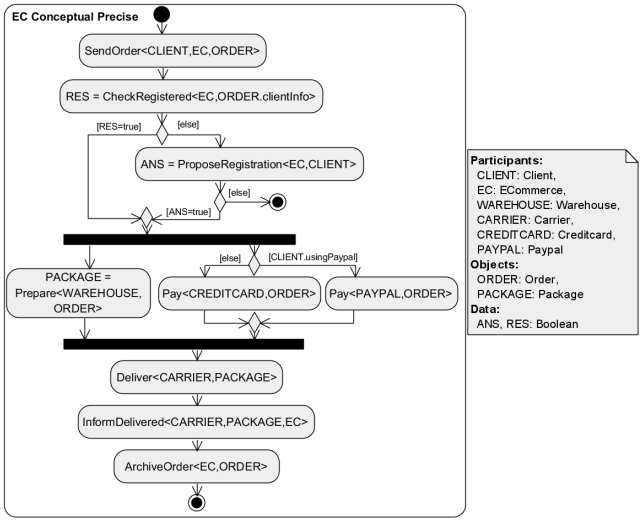
b) The participants will have a name and will be typed by a class with stereotype either ≪businessWorker≫ or ≪system≫, the objects also will have a name and will be typed by a class stereotyped by ≪object≫, and the data will be typed by UML datatypes, either predefined or user defined in the static view. It is possible to constrain the possible participants, objects and data of a business process.

c) The fact that some participants and objects of the business process take part in a task is modelled by means of dependency relations linking the task class with the participant/object classes. The task classes should depend on the object classes (to depict that the tasks will act over them), and the participant classes should depend on the task class (to depict that they will take part in the tasks). Furthermore, the dependency between a task and an object class may be stereotyped by ≪out≫ if such business object is created during the task. If a task is characterized by some data, then those data are represented by attributes of the task class typed by datatypes. An attribute may be stereotyped by ≪out≫, in the case of a data produced by the task itself. Some constraints of kind invariant/pre/post may be attached to a ≪task≫ class expressing relationships holding always during/before/after the task execution among the involved entities. The behavioural aspects of a task instead may be modelled using some of the many constructs offered by the UML, e.g., sequence and again activity diagrams. The action nodes of the activity diagram modelling a business process will be labelled by instances of task classes, presented in the following way:

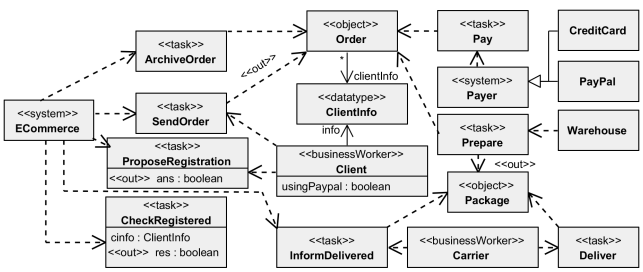
TaskName<x_1,...,x_n>

where TaskName is a task class, and x_1,...,x_n are the participants/objects/data involved in the task.

Fig. 4 shows the model of EC following the Precise Conceptual style. In this case we have put together the static and the task view and thus the model consists of a class diagram, an activity diagram and the participant/object/data lists.



(a) Behaviour View (Activity Diagram)



(b) Static and Task Views (Class Diagram)

Figure 4: EC: Precise Conceptual Model

The class diagram introduces the classes defining the participants and the objects, together with some datatypes used to describe them (for example ClientInfo). EC, PAYPAL and CREDITCARD are participants of kind system (they correspond respectively to the software system running the e-commerce site, the Paypal payment service and the credit card handling system), whereas CLIENT is a human participant and CARRIER and WAREHOUSE are respectively an external transport company and a department of the e-commerce company; they are not classified as systems since they cannot be fully automatized. Notice that the client is not involved in the task for delivering the package because it is assumed that the delivery will be made at a certain address and does not require an active participation of the client itself. Instead, the task informDelivered involves two participants, the CARRIER and the EC system.

2.5 The Precise Style "Operation Based"

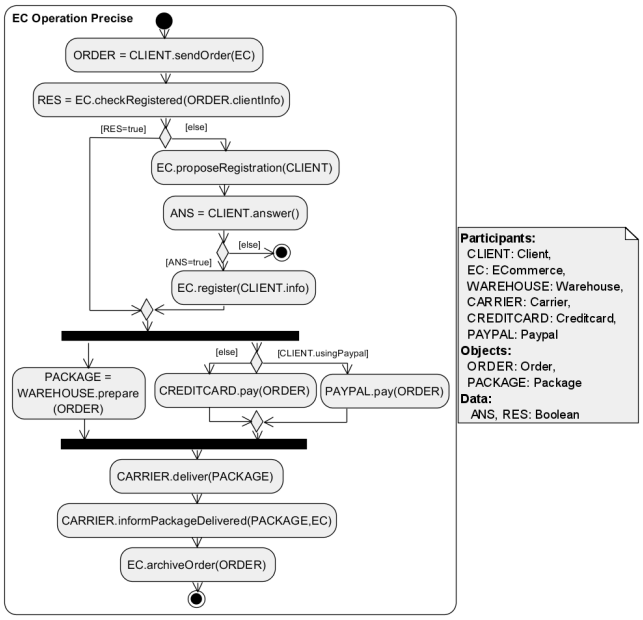
The *Precise style "Operation Based"* (shortly *Precise Operational style*) for BP modelling requires to describe the participants, the objects and the data precisely by means of a static view, as for the Precise Conceptual style of Sect. 2.4, and similarly the activity diagram modelling the behaviour of the process is presented in a precise way. The only difference concerns the way the UML is used to model the tasks: now the tasks are modelled by means of calls of operations of the participant or object classes. Thus, the Precise Operational model of a business process consists of (see Fig. 1): *i)* a static view, *ii)* the participant/object/data lists, *iii)* and an activity diagram representing its behaviour satisfying the following constraints.

The tasks involving the participants and the objects will be modelled by operations of the various participant/object classes stereotyped by ≪task≫ (whenever all the operations of a class have this stereotype it may be omitted to make the visual presentation simpler). When defining the ≪task≫ operations, it is important to keep in mind that *i)* an operation corresponding to a task part of a class C stereotyped by ≪businessWorker≫ or ≪system≫ describes a task that a participant of type C is responsible to perform and it should be named using a verb in the infinitive form without the "to"; *ii)* an operation corresponding to a task part of a class C stereotyped by ≪object≫ describes a tasks that will be done over an object of type C and it should be named by the past participle of a verb.

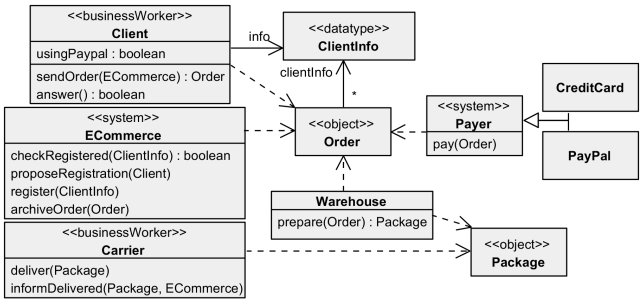
The action nodes of the activity diagram will be decorated either by calls of task operations on either participants or objects (and the participants, the objects and the data will freely appear as arguments of them) or by UML actions, i.e., assignment, creation and destruction of class instances; whereas the conditions on the edges leaving the decision nodes will be OCL expressions in which the participants, the objects and the data will freely appear.

Notice how the guidelines asking to define task operation either corresponding to operations over objects or to something that a participant is responsible to execute leads to have fine-grained tasks more or less of the same size, and finer than those of the Precise Conceptual style.

Fig. 5 presents the Precise Operational model of the EC business process consisting of a class diagram, an activity diagram, and the typed participant/object/data lists. The class diagram introduces the classes defining the participants/objects/data as for the Precise Conceptual model of



(a) Behaviour View (Activity Diagram)



(b) Static View (Class Diagram)

Figure 5: EC: Precise Operational Model

Sect. 2.4. Since all the operations are stereotyped by ≪task≫ we have omitted to depict it visually in the class diagram. Notice that in this model the request for registration to the client is implemented by three tasks: the request by EC, the client answer and then the registration of the same by EC, in case of positive answer; and thus differently than in the Precise Conceptual and in the Disciplined models. The reason is the style guideline asking to attach to each participant the tasks which are under his responsibility, and so we had to single-out the answering of the client.

In this example there are no task operations on the classes modelling the business process objects, because we have modelled explicitly who is performing the various tasks; instead, if we would prefer avoiding to express who is delivering the package, we can delete the class Carrier and add two operations to the class Package (requiredDelivering and delivered).

3. CHOOSE THE MOST SUITABLE STYLE

Even if the precise styles are better for what concerns the expressiveness and the quality of models, as shown in [6, 7], we think that the context in which the business process models appear (the why, when, where, how long, by whom the models are produced/used) and the wanted quality of the same models may influence the choice of the most suitable style. Hence, we have devised a method for choosing the most suitable style trying to balance precision and freedom, similarly to the proposal of [2] for choosing between agile and prescriptive software development methods.

	Factors	Weights		Influence: Ultra-Light	Light	Disciplined	Precise Conceptual	Precise Operational		Modelling Context	
F1	Modeller has a good knowledge of UML	W1	0.8	I1	-2	-1	0	1	1	MC1	[-1,0,1]
F2	Model reader has a good knowledge of UML	W2	0.8	I2	-2	-2	-2	1	1	MC2	[-1,0,1]
F3	Model used as a communication media	W3	0.8	I3	0	1	2	1	-1	MC3	[0,1]
F4	Model used in a MD Development as source of (semi-)automatic transformations	W4	0.8	I4	-2	-2	-1	1	2	MC4	[0,1]
F5	Model must be produced in a short time	W5	0.8	I5	2	1	0	-1	-2	MC5	[0,1]
F6	Model will have a long life span (and perhaps evolve)	W6	0.5	I6	-2	-1	1	1	1	MC6	[0,1]
F7	Modelled business is critical	W7	1.0	I7	-2	-1	0	2	2	MC7	[0,1]
F8	Model quality should be checkable	W8	0.5	I8	-2	-1	0	2	2	MC8	[0,1]

Figure 6: Factors and weights for choosing the most suitable style

To choose the most suitable style, we have created an algorithmic *suitability* estimation method. Our method allows the user to obtain a *suitability value* in a given context, for each style proposed in this paper, simply assessing the presence, in such context, of some factors, listed in Fig. 6. There is a group of factors concerning the environment in which the model will be developed and used (F1,..., F7), and a factor concerning the possibility to check the quality of the model itself (F8). These factors have been selected empirically[5]: the authors have independently proposed a list of factors, prompted by their (long for some of them) modelling experience in many different contexts and producing models of different quality; then such lists have been compared, discussed and then merged in a "not too long" common list (the considered factors were present in all the author lists).
A user of our method has just to decide which factors are present in her/his Modelling Context (MC) and return the numbers MC_1,...,MC_8 in the following way:

– for i = 1,2: if the factor F_i is present, then $MC_i = 1$;
if F_i is not present, then $MC_i = -1$;
if F_i is not relevant/known $MC_i = 0$;
– for i = 3,...,8: if the factor F_i is present, then $MC_i = 1$,
else $MC_i = 0$;

Then, the formula (**) will assign a suitability value to each style S and the style with the higher value will be considered the most suitable.

(**) Suitability of Style S: $ST_s = \sum_{i=1,...,8} W_i * I_{i,s} * MC_i$
where $S \in$ {Ultra-Light, Light, Discip., Prec. Conc., Prec. Oper.}
– $I_{i,s}$ represents the influence of factor F_i on the suitability of the style S: –2 when it is heavily deterring it and 2 when it is heavily favouring it: $-2 \leq I_{i,s} \leq 2$;
– W_i represents the importance of factor F_i: near 0 when the importance is negligible and 1 when it is of paramount importance: $0 < W_i \leq 1$.

To guarantee a sensible selection of the used weights (W_i, $I_{i,s}$, $i = 1, ..., 8$), the authors have empirically[5] estimated them. More in detail, all the authors have independently proposed some values for them, and then such values have been compared and discussed. In the cases of disagreement (few and where the difference was not excessive) a common value has been decided. The values of the various weights (W_i, $I_{i,s}$, $i = 1, ..., 8$) can be seen in Fig. 6.

[5]originating in or based on observation or experience (Merriam Webster's Dictionary)

	F1 (MC1)	F2 (MC2)	F3 (MC3)	F4 (MC4)	F5 (MC5)	F6 (MC6)	F7 (MC7)	F8 (MC8)	Ultra-Light	Light	Discip.	Prec. Conc.	Prec. Oper.
A)	-1	-1	1	0	1	0	0	0	**+4,8**	+4,0	+3,2	−1,6	−4,0
B)	1	1	0	1	0	1	1	1	−8,8	−6,0	−1,9	+5,9	**+6,7**
C)	1	-1	1	0	0	1	1	0	−3,0	+0,1	**+3,7**	+3,3	+1,7

Figure 7: Applications of the style suitability evaluation method

An Excel spread sheet freely available[6] allows to easily insert the values characterizing the presence of the various factors (MC_i, $i = 1, ..., 8$) and will do all the needed calculations, presenting the suitability values for the five styles (so the weights W_i, $I_{i,s}$, are reported here to give a complete explanation of how the style selection method works; the user have only to assess the presence of the various factors just filling the gray column in Fig. 6).

Let us discuss the various factors and their impact on the choice of the most suitable style, as formalized by the chosen weights. F1 and F2 take into account the knowledge of the UML by who produces and reads the model (notice how the three light styles are equally suitable for readers not fluent in the UML notation: since all of them require just the knowledge of the visual constructs of the activity diagrams). F3 and F4 concern the role of the produced model, and we consider two dimensions: whether it is relevant that the model is easy to read, because it is a piece of documentation, penalizing the choice of the precise-styles and heavily-favouring the Disciplined style, and whether it has to be used as the starting point of some (semi-)automatic transformations, favouring the precise styles (mainly the Precise Operational). Then, we consider whether it is important to produce the model in a short time (F5), since the time obviously increases as the style becomes more precise. Factor F6 is related to the span life of the model and thus to the possibility to undergo some evolution, which equally favours the Disciplined and precise styles (indeed, the structuring of the textual decoration of the Disciplined activity diagram is a great support to evolution, think, e.g., to replace a participant with another one, in this case it will be easy to find all the tasks in which s(he) was taking part). F7 is the most influential factor for the choice of a precise style, and again heavily favours the precise styles (in this case in the same way, since they differ only in the level of abstraction, not in rigor). F8 concerns the possibility to check the quality of the model, i.e.: are you interested in checking if (1) is the model complete and minimal and without UML errors? (2) are rigorous inspections on the model doable? (3) are complexity metrics on the model definable?

We have then validated these weights by applying the evaluation method to many cases out of the possible $576 = 3^2 * 2^6$ (several cases are reported in the Excel spreadsheet). Here we report three typical modelling contexts: **A)** a draft model of a business process to be discussed with the stakeholders; **B)** a model to be used as the starting point of the (semi-)automatic generation of a BPEL implementation of a system supporting a business process; and **C)** a model of the process for getting the approval of a new building project by the local administrations in Italy. In Fig. 7, we present the values characterizing the three cases and the evaluation results. We got that for case **A)** the Ultra-Light style is the most suitable (+4,8), for case **B)** the winner style is the Precise Operational (+6,7), and, finally, for case **C)** the winner is the Disciplined style (+3,7). These results are quite reasonable: **A)** the Ultra-Light model is acceptable since it will be just used to discuss on the process with stakeholder, and will be heavily modified before to reach a stable form; **B)** the Precise Operational is the most suitable style to get a model to transform into a running system using BPEL; **C)** the Disciplined style allows a quite precise description of the things to be done and the item to be handled but at a quite abstract level and in a way easy to read for non-technical persons, not cluttering the model with too many details, but at the same time helping to avoid mistakes that may have very serious consequences (e.g., to name in the same way two slightly different documents or tasks).

[6] `http://softeng.disi.unige.it/TR/Styles.xls`

4. CONCLUSION AND FUTURE WORK

In this paper, we have first presented five styles, differing in the precision level, for modelling the business processes by means of UML activity diagrams. Then, we have devised a practical empirical approach to choose a style among the proposed ones, keeping in consideration the context in which they will be used and for what.

In this work, we focused on business process modelling and chose UML activity diagrams for their representation. However, we think that our work (styles and our method for the choice of the style) can be generalized for other UML diagrams, notations, purposes and in other contexts/settings with some rework. For example, a similar proposal could be put forward for UML state machines used to describe the behaviour of an entity or for business processes expressed by means of BPMN.

As future work, we would like to test more systematically and deeply our practical approach. In particular, we intend to better validate our weights with additional cases (i.e., modelling contexts) and try our approach in an industrial context. Another interesting direction could be re-thinking the approach using machine learning techniques.

5. REFERENCES

[1] S. W. Ambler. *The Elements of UML 2.0 Style*. Cambridge University Press, 2005.

[2] B.W. Boehm and R. Turner. *Balancing agility and discipline: a guide for the perplexed*. Addison-Wesley.

[3] E. Di Nitto, L. Lavazza, M. Schiavoni, E. Tracanella, and M. Trombetta. Deriving executable process descriptions from UML. In *Proceedings of ICSE 2002*, pages 155–165. ACM, 2002.

[4] M. Fowler and S. Kendall. *UML Distilled: A Brief Guide to the Standard Object Modeling Language (4th Edition)*. Addison-Wesley Professional, 2010.

[5] S. Jurack, L. Lambers, K. Mehner, G. Taentzer, and G. Wierse. Object flow definition for refined activity diagrams. In *Proceedings of FASE 2009*, volume 5503 of *LNCS*, pages 49–63. Springer, 2009.

[6] G. Reggio, M. Leotta, and F. Ricca. "Precise is better than Light" A Document Analysis Study about Quality of Business Process Models. In *Proceedings of EmpiRE 2011*, pages 61–68. IEEE, 2011.

[7] G. Reggio, F. Ricca, G. Scanniello, F. Di Cerbo, and G. Dodero. A precise style for business process modelling: Results from two controlled experiments. In *Proceedings of MODELS 2011*, volume 6981 of *LNCS*, pages 138–152. Springer, 2011.