A Modeling Approach for Bioinformatics Workflows

Laiz Heckmann Barbalho de Figueroa, Rema Salman, Jennifer Horkoff, Soni Chauhan, Marcela Davila, Francisco Gomes de Oliveira Neto, Alexander Schliep

A Modeling Approach for Bioinformatics Workflows

Laiz Heckmann Barbalho de Figueroa[1], Rema Salman[1], Jennifer Horkoff[1,2], Soni Chauhan[1], Marcela Davila[2], Francisco Gomes de Oliveira Neto[1,2], and Alexander Schliep[1,2]

[1] University of Gothenburg, Gothenburg, Sweden
[2] Chalmers University of Technology, Gothenburg, Sweden
gushecla,gussalmre@student.gu.se, jennifer.horkoff@gu.cse.se

Abstract. Bioinformaticians execute frequent, complex, manual and semi-scripted workflows to process data. There are many tools to manage and conduct these workflows, but there is no domain-specific way to textually and diagrammatically document them. Consequently, we create methods for modeling bioinformatics workflows. Specifically, we extend the Unified Modeling Language (UML) Activity Diagram to the bioinformatics domain by including domain-specific concepts and notations. Additionally, a template was created to document the same concepts in a text format. A design science methodology was followed, where four iterations with seven domain experts tailored the artefacts, extending concepts and improving usability, terminology, and notations. The UML extension received a positive evaluation from bioinformaticians. However, the written template was rejected due to the amount of text and complexity.

Keywords: UML, activity diagram, workflow, bioinformatics.

1 Introduction

Bioinformatics is a branch of biology, which is connected to computational methods for biological data generation. Generating data for biological analysis, such as DNA sequencing, requires several connected tools in a workflow, defined as a sequence of tasks that cover the steps of a process from initialisation to producing final results [10]. Bioinformaticians create workflows that need to be followed precisely to achieve satisfactory results [13]. To design and manage these workflows, bioinformaticians use a mixture of tools and frameworks from various sources [10,2], often interspersed with manual steps and checks.

Work in [2] reported usability challenges when using available tools, such as limitations on data visualisation and patterns for workflows. Additionally, [11] describes the lack of features, notations, or concepts, such as the absence of loops. Our experience with a local bioinformatics lab reveal that workflows are incredibly complex, often implicit, and involve decisions without clear-cut criteria. These limitations hinder bioinformaticians and researchers in visualizing, sharing, replicating and improving workflows.

The literature reports languages used to describe bioinformatics workflows, e.g., Domain-Specific Languages (DSLs) can be tailored to bioinformatics [5]. UML has been adapted for bioinformatics processes (e.g., [19]). However, this work does not focus specifically on capturing manual and scripted bioinformatics workflows and does not address the issues identified above.

The purpose of this study is to create a usable modeling language for capturing and understanding bioinformatics workflows. The long-term aim is to establish a shared understanding and consistency between the activities of the involved parties; create sharable documentation to provide a clear vision of the process; support training new bioinformaticians; identify problems in the workflow design; reduce the bioinformatician's reliance on individual interpretation; increase the replication precision of the analysis; and improve traceability. To develop and evaluate our solutions, we have worked with bioinformaticians at the University of Gothenburg's Bioinformatics Core facility[3], following a Design Science Research Methodology (DSRM), to answer the main research question and its three sub-questions:
RQ1: How can we support modeling of bioinformatics workflows in an effective and usable way?

- RQ1.1: What are the defining and unique characteristics of bioinformatics workflows compared to standard workflows?
- RQ1.2: How should workflows, including the concepts discovered in RQ1.1 be visualised to be understandable by the bioinformaticians?
- RQ 1.3: How can we design a useful and understandable template to document the concepts from RQ1.1?

The rest of this document is structured as follows: Sec. 2 describes how DSRM was used to develop the artefacts. Sec. 3 presents the final artefacts and the results for each iteration, while Sec. 4 discusses the findings and limitations. Sec. 5 compares with related work, while Sec. 6 concludes the paper.

2 Methodology

This paper uses the DSRM due to its pragmatic nature and strength in solving real-world problems [9]. The DSRM procedure proposed by Peffers et al. in [17] was adapted to the needs of this research, as summarized in Fig. 1. Based on the problems identified in the 0^{th} iteration, three artefacts were created, evaluated, and improved: the UML Activity Diagram (AD) meta-model extension, its concrete syntax, and the Workflow Description Specification Template (WDST).

Facilities: The research was conducted with participants from three different facilities: the Bioinformatics Core Facility, part of the Sahlgrenska Academy Core Facilities at the University of Gothenburg; the Genomic Medicine Sweden (GMS); and the Translational Genomics Platform[4].

[3] https://cf.gu.se/english/bioinformatics
[4] https://wcmtm.gu.se/research-groups/genomics-platform

Participants: The head of the Bioinformatics Core Facility (the 5th author) used a purposive sampling technique to select the participants for this research. This technique aims to diminish the accidental sampling bias as the participants' selection is based on the researchers' belief that they fulfil stipulated criteria [25], in this case, workflow knowledge. The seven participants are identified as P1 to P7. The four DSRM iterations are briefly described below.

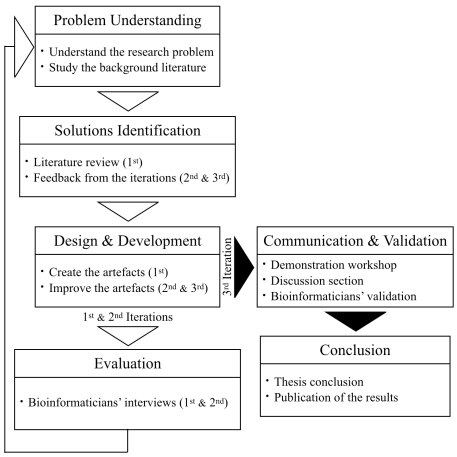


Fig. 1. The DSRM Process in this Study.

0^{th} Iteration. In the first exploratory iteration, two of the authors (MSc. student and a modelling expert) worked iteratively with the head of the Bioinformatics Core Facility to map out 2–3 specific workflows, as initial exploratory examples. Challenges and concepts specific to bioinformatics were noted, feeding into the next rounds.

First to Third Iterations. Based on the findings from the 0^{th} iteration, as well as ideas from the literature, we created three artefacts and evaluated them with the seven bioinformaticians (P1–P7). During the first and second iterations, we conducted semistructured interviews that lasted a maximum of, respectively, 30 and 60 minutes with five bioinformaticians each. The interviews were hosted at the laboratory's facility, recorded upon interviewees' agreement, with assured anonymity of the participants' answers. All interview questions and other materials for the study can be found in [6].

During the interviews for the first iteration, the created WDST, two concrete syntaxes, and two examples were presented, eliciting opinions via the pre-set questions. In the second iteration, participants were asked to draw for 15 minutes a workflow of their choosing using the updated notation by using a stencil in https://www.draw.io/. They were also asked to fill in a WDST template for 15 minutes in Google sheets. When the participants were using the artefacts, they were asked to follow the think-aloud protocol [7], while the observations were recorded in a log template by a researcher. In the end, they answered questions about language and method usability, inspired by the System Usability Scale (SUS), a widely used ten-item survey to assess usability and learnability [4].

In the final iteration, all participants from previous iterations were invited to the one-hour workshop, recorded upon their approval. In the workshop, the artefacts were described through examples and participants were paired to discuss the usability and understandability of the notations and concepts. After that, each pair explained their thoughts, and then the participants individually and anonymously validated the notations and concepts using a survey via Mentimeter[5]. All workshop material can be found in [6].

[5] https://www.mentimeter.com

After transcribing the data we conducted thematic analysis to identify significant patterns, grouping them into themes [20]. After coding, the suggestions and problems were addressed during the *Solutions Identification* and *Design and Development* steps in each iteration. In the last iteration, the artefacts were not further refined; with the changes suggested for future work.

3 Results

In this section we will present the final artefacts and the output from each iteration. Note that, for space considerations, we present only the final artefacts, but their intermediate versions for each iteration can be found in [6]. Below, in each iteration, we describe (Fig. 1): a starting point (i.e., Solution identification); the work on the extension of the AD meta-model, concrete syntax and the WDST (Design & Development), and an Evaluation performed. Note that the evaluation reveals suggestions and solutions taken as the starting point of the upcoming phases.

3.1 Final Artefacts

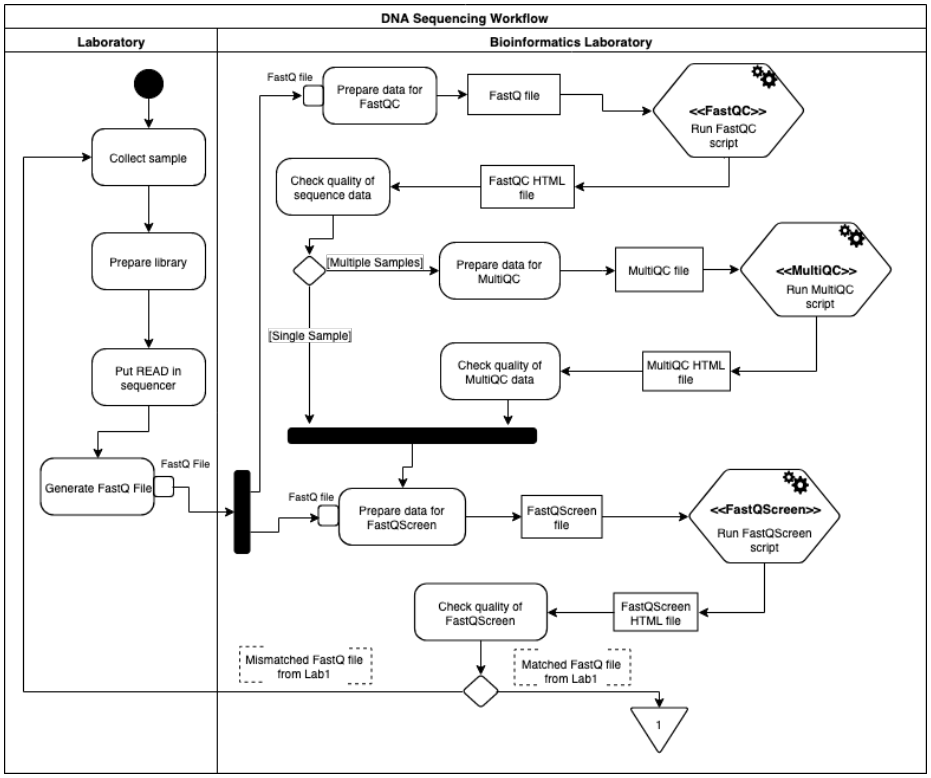


Fig. 2. A Bioinformatics Workflow Example using the Final Version of the Language.

We show a small example using the final version of the language, used in the evaluation workshop (Fig. 2). The final version of the developed artefacts includes a UML Activity Diagram (AD) meta-model extension for bioinformatics domain (Fig. 3); an excerpt of the final version of the WDST (Fig. 4); and the final concrete syntax (Table 1). The following sections describe the iterative results that lead to these artefacts.

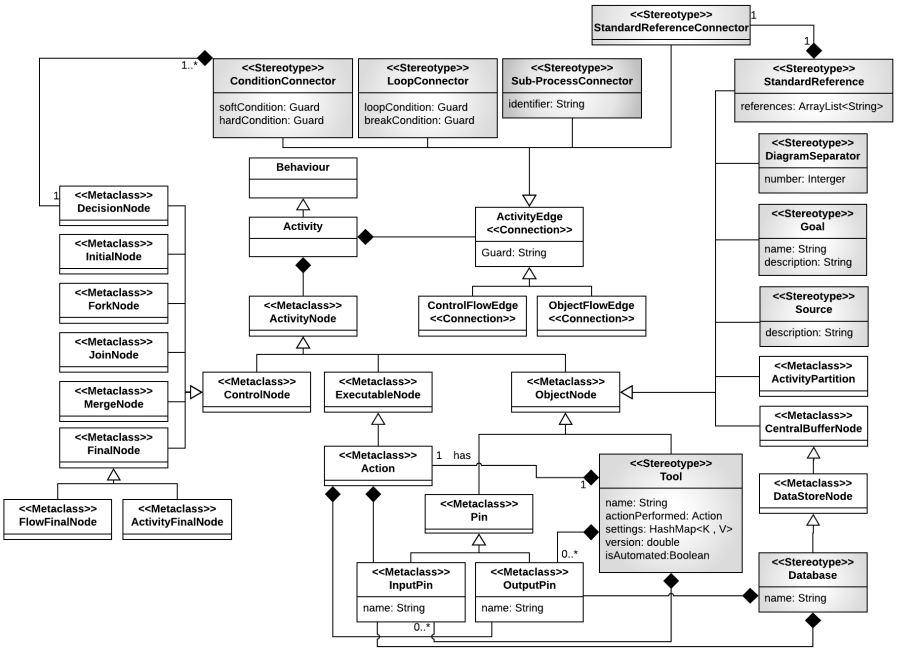


Fig. 3. The Final Version of the Extended UML AD Meta-Model (white classes are from UML AD [16], while grey classes were added in this work).

Workflow Description Specification		
Workflow ID:		*<<the workflow name or identifier>>*
Date of creation: *<<date in which this document was created>>*		Number of steps: *<<amount of steps>>*
Workflow version: *<<version of this document>>*	Modification date: *<<date of modification>>*	Workflow creator: *<<name>>*

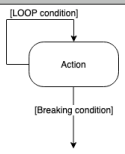
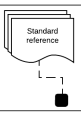
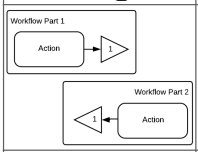
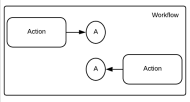
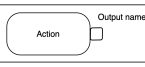
Workflow	
Workflow goal:	*<<what do you want to achieve with this workflow?>>*
Workflow source:	*<< Is this workflow created locally? or it follows a reference - in that case, add link to the reference or name the person>>*
Workflow responsible:	*<<person who signs the final output or who uses this workflow>>*

First Step (Start point)	
Step ID:	*<<The name or identifier of the start step>>*

Final Step (End point)	
Step ID:	*<<The name or identifier of the start step>>*

Fig. 4. An Excerpt of the Final Version of the Workflow Description Specification Template - WDST.

Table 1. The Final Meta-Model Concepts and the Sources of each Concrete Syntax with their Notations and Explanations.

Concept Name	Concrete Syntax Source	Notation	Explanation
Loop	UML structured nodes [23]		Follow the *Loop* semantics and syntax suggested for UML, where using arrows with guards lead to activity repetition. Helps to capture complex and repetitive tasks found in bioinformatic workflow examples.
SoftCondition	UML AD [16] & different usage of line styles from [1]		Follows the standard UML AD semantics and usage, where the guard syntax was changed to dashed lines. It is used to captures fuzzy thresholds.
HardCondition			Follow notation and concept of the guard in the standard UML AD. It captures hard thresholds as found in practice.
Source	Flowchart notations & i* visual syntax [15]		Concept identical to *Resource* in i*, using the document notation from the flowchart notations. Captures source of task or action, often a research paper or external reference.
Tool (Manual)	Flowchart notations & i* visual syntax [15]		The task concept is from i* visual syntax with an additional icon on its corner to allow a faster visualisation of the tools, depending on the mode (manual or automated). It captures tasks operated by or through tools.
Tool (Automated)			
Database	UML AD extensions in [22]		Concept identical to UML *Datastore*, but with the flowchart cylinder shape, *Database* notation. It captures storage of files.
StandardReference-Connector	UML AD notes connector [16]		Connects between the *StandardReference* notation and its *InputPin*.
StandardReference	UML AD [16]		To add standard data as input to be compared with the data being analysed, differentiating them from the ordinary input. It is used to show that this data is not part of the data flow.
DiagramSeparator			The semantic and syntax are inspired by *ActivityEdgeConnector* with a graphical modification, a triangle with a number instead of circles with letters. Helps to deal with large workflows via diagram splitting.
Sub-processConnector			Identical to the semantic and syntax of UML AD *ActivityEdgeConnector* with a different name. Help to compress parts of the workflows.
OutputPin			Follow exactly the standard notations and usage in UML AD. The standalone pin is the same file between two consecutive steps. Input and output are often file exchanges in bioinformatics workflows. Helps to show the data flow.
InputPin			
StandalonePin			

3.2 0th Iteration

In the first exploratory iteration, we attempted to capture examples of workflows in several existing modeling languages, including Business Process Model and Notation (BPMN) and Data-Flow Diagrams (DFDs). We found BPMN to be too complex for our purposes, for example, we did not make use of most different types of gateways. Given that the target end users were not native modelers, we perceived AD to be simpler to build on. We also found it easier to express the flow of file inputs and outputs in AD, although this is also possible in BPMN. Finally, we made extensive use of conditional forks and joins, and we found the visual guard condition ([condition]) in AD quite convenient for this. We found that DFDs were limited in capturing the usage of tools in the workflow, a key element for bioinformaticians. In the end, we settled on UML Activity Diagrams as they: encompass an appropriate level of complexity, support extensibility, come with familiarity (for IT specialists), and the support of the UML community [16].

Our early examples revealed gaps in AD, which motivated further iterations. We found that bioinformatics workflows involve: 1) many complex and repetitive tasks; 2) many 'quality checks' of tool outputs using threshold values which could sometimes be subjective to interpretation; 3) constant splitting of tasks between people and tools; 4) data emphasis where files were exchanged back and forth; and 5) unclear motivation behind some tasks.

We also created a draft template aimed to help elicit and capture bioinformatics workflows. The idea was that bioinformaticians are not necessarily experts in structured modeling languages, such as UML. Therefore, they may be more comfortable capturing process details in text via a template. These findings and the first draft of the template were used as input to the next iterations.

3.3 First Iteration:

Solutions Identification: The 0^{th} iteration identified *thresholds*, *source*, differentiation of *files*, *goals*, *sub-process*, and *repeated iterations* as needed by bioinformaticians while creating their workflows. We started by incorporating each of those concepts into our three artefacts.

AD meta-model extension: Our starting point was the UML AD metamodel in [16]. Based on the nature of the UML profile, all the UML default AD syntax and semantics were kept (e.g., action, decision, join, forks). Additionally, the concepts *activityPartition* (swimlanes) and *activityEdgeConnector* from the UML AD [16] maintained the same syntax, where the former was based on [21] and the latter was used to represent sub-processes for the bioinformatics domain (a connection point instead of drawing a long process).

The implemented extensions included the added stereotypes: *tool*, *diagramSeparator*, *source*, and *goal*, which were inherited from the meta-class *objectNode* classifiers. The *tool* has a composition relationship with the meta-classes *action*, *inputPin*, and *outputPin*. Due to some changes on the *datastore* notation, the class was added as a stereotype. Additionally, the *loopConnector* was inspired by [23], inherited from the super-class *activityEdge*, containing *loopCondition* and *breakCondition* guards, and *thresholdConnector* inherited from

the super-class *activityEdge*, containing the specified guards *softThreshold* and *hardThreshold*. The *decisionNode* composites at least one *thresholdConnector*.

Concrete Syntax: The design decisions for our concrete syntax, considered the principles for cognitive effectiveness of the visual notations, which are: symbols deficit, redundancy, overload, and excess. These principles ensure the correspondence between semantics and graphical shapes of notations [14,15], which is part of the Visual Alphabet theory and Physics of Notations theory. We followed the UML AD patterns, while avoiding to use different colours or texture to define the visual syntax of the concepts. This results in an inclusive language that can be used by any person with visual disabilities or colour blindness.

WDST: We added workflow information to the WDST, such as *workflow* and *step ID*, *name*, *creator*, *version number*, and *date of creation*.

Evaluation: Five bioinformaticians (P1–P5) from the three facilities were interviewed to evaluate the WDST. The diagram users should be bioinformaticians and stakeholders. The results for this first iteration included: improving the *understandability* use of the *swimlanes* and *loop*s inclusion and exclusion factors; the addition of a *tool settings and parameters* field (by two participants); three participants outlined the notations usage as *system documentation*, thoughts *structuralisation*, and process overview; the current state of the workflow diagrams depends on an individuals' drawing style. Moreover, all participants said that they would draw the workflow first and then fill the WDST. However, P2 stated that "I think like there's so much here (WDST) that would be redundant when you're using this (both artefacts)".

3.4 Second Iteration

Solutions Identification: The solutions for this iteration come from the participants during the interviews in the previous iteration (i.e., the results described in the previous evaluation).

AD meta-model extension: We included more attributes to the stereotypes *tool*, *standardReference*, and *threshold*. We renamed the *threshold* and the meta-classes *activityEdgeConnector* and *datastore*. We added stereotype *standardReferenceConnector* as an inheriting classifier of *activityEdge* because the *standardReference* was mentioned as missing by two participants. However, based on participant feedback, we modified the naming of *thresholdConnector* and *datastore* to *conditionConnector* and *database* respectively, as visible in the final version of the meta-model (Fig. 4).

Concrete syntax: We only added a sub-concept for input, the *standardReference* (see Table 1 for its design sources and explanations).

WDST: Based on the participants' feedback, we added, reworded, and deleted repetitive and unnecessary fields. The goal was to decrease redundancy and increase familiarity. We added guidance for the template usage and the required input and output data for each tool, as well as more information about their version and settings. We also added 'conditions' to the 'thresholds' section. Additionally, participants requested to change the role of the WDST to become a

standardised way to document workflows for stakeholders and to share knowledge, as opposed to a simple helper during the workflow elicitation process.

Evaluation: Five participants (P1, P3–P6) were interviewed and recorded. Regarding the concrete syntax, Two participants indicated that the *goal* notation was *unneeded* and two other participants pointed *vertical* and *horizontal join/fork* as *unfamiliar*. Two participants requested to *add*: i) the parallelogram shape of pins, ii) no database with in/output pins, and iii) different arrow shapes. Participants indicated the provided stencil of workflow shapes would be used, but not frequently since it is time-consuming to draw workflow diagrams, usually created only for publications. Even though the notations complexity was considered low by four participants, the other two participants stated that the number of graphical shapes was high. The participants suggested a descriptive *manual* to guide the users, while others stated that *training* is necessary. Finally, the participants felt confident using the notation stencil but highlighted challenges when using `draw.io` as a modeling tool.

Regarding evaluation of the WDST, the participants indicated that its content flow was good. However, three of the participants stated that they would not use the template because of its complexity (i.e., the amount of information to be written) and time consumption. The participants mentioned that *training* (e.g., a user manual or usage examples) should help the users. Conversely, two participants said that the WDST grey-text is sufficient and self-explanatory.

In summary, participants' general impressions of the artefacts were that the diagram is good, useful, and provides a clear overview, whereas the WDST requires time and holds much information. Additionally, P4 stated that both artefacts "complement each other."

3.5 Third Iteration

Identify Solutions: Similarly to the second iteration, the solutions for this iteration come from the previous iteration's evaluation.

UML AD meta-model extension: We added a composition association between the *database* stereotype class and the *input* and *output pins* meta-classes. Additionally, we added an attribute to the *tool* stereotype-class to identify if the tool is automatically or manually operated (Fig. 3 shows the updated meta-model).

Concrete syntax: Improvements to the concrete syntax include: i) changing the location of the *inputPin* on *tool* to ensure the vertical gradient of the diagram, ii) attaching *inputPin* and *outputPin* to the *database* to represent the data flow and keep the consistency between shapes in the XML notations stencil, iii) improving the *action* and *tool* descriptions to decrease confusion, iv) adding a separate text field for the performed activity on the *tool* shape to remove the issue of deleting the name or performed activity when writing them, v) adding a new notation for the manually operated *tool* to increase transparency of the automation level, vi) removing the *goal* notation upon participants' request, and vii) adding the *standalonePin* to the stencil to include familiar notations to the bioinformaticians. See Table 1 for the final version of the concrete syntax.

WDST: The WDST annoyed the participants because of its documentation traceability fields and its descriptive nature, which was unfamiliar to the participants. Some of the changes implemented were: further explanation for several cells by using a light grey text, format fixing on the cell *tool settings and parameters*, removal of the *workflow name* due to its interchangeable use with *workflow ID* by the participants, and the word 'process' in the sentence *process step* from the WDST, addition of a basic excel formula to linking the *workflow ID* on the first page to the second page to avoid typing the same information twice, and a conditional formatting that changes the text colour while filling cells from grey to black. The grey text fields held the explanation to help new users and were thus kept since they are vital to the WDST understandability.

Evaluation: Six participants (P1–P3, P5–P7), including the head of the Bioinformatics Core Facility, joined the workshop to evaluate the final version of the artefacts. Regarding the concrete syntax, participants' feedback revealed that the notations and concepts are understandable and simple, but they requested improvements related to better concepts definition and different software for drawing the diagrams. One participant wanted the diagrams to be automatically generated, as in Snakemake (https://snakemake.readthedocs.io). The bioinformaticians outlined *fork nodes*, *join nodes*, *swimlanes*, and *standardReference* as unnecessary notations. Additionally, they said that the diagrams would be used for final and standard documentation, after sketching, and stated that the notations would increase the time spent to draw the current workflows, which were described as having overloaded and overused *boxes*, and *notes* symbols.

Table 2. Mentimeter validation results in the third iteration.

Question	Median	Mean
How understandable are the presented concepts and notations?	3	4.3
How easy is to use the concepts and notations library?	3	3.7
How likely would you use the concepts and notations in a diagram?	3	3
How likely do you believe a stakeholder can understand the concepts and notations?	3	2.8
How understandable is the documentation for you?	3	2
How easy is to fill the documentation template?	3	1.7
How likely would you use the documentation template?	3	1.3
How likely do you believe a stakeholder can understand the documentation template?	3	1

The participants answered Likert scales and an open-ended question for each artefact using Mentimeter, see Table 2 for the results with mean and median values. Here, 1 is very unlikely, incomprehensible, or arduous, while 5 is very likely, understandable, or easy. The results show that the participants find the concepts and notations of the stencil understandable with an average of 4.3, where 3.7 reflected ease of use. The participants would likely use the concepts

and notations; with an average of 3, and 2.8 is their average perception of stakeholders' understandability. Nevertheless, the open-ended question had similar results as the qualitative workshop results. However, one participant requested a further improvement to, "make it easier to add several outputs". Moreover, a participant proposed renaming the *soft-condition* to "manual-inspection or manual evaluation" and changing its concrete syntax to differentiate it even more from the *hard-condition*. A participant abstained from answering.

Regarding evaluation of the WDST, the participants *disliked* the amount of typing, identified traceability issues, and mentioned that the stakeholders could have trouble understanding the WDST because of its complexity. They also outlined that *automation* would save time when producing the written template from graphs since the WDST was indicated as time-consuming. Table 2 shows that the WDST was deemed incomprehensible by most of the participants, with an average of 2 and 1.7 regarding the ease of filling it. The participants would be very unlikely to use the WDST ($\mu = 1.3$) and they do not believe that the stakeholders would understand it ($\mu = 1$). Regarding the open-ended question, five participants agreed that it is complicated. Thus, they suggested simplifying it by removing most of its content, keeping only the *tool section*, and adding a place to input the command line commands. One participant left the question unanswered.

4 Discussion

Here, we return to answer our research questions by summarizing the main results found throughout all iterations, followed by the limitations of our study.

RQ1.1: The defining and unique characteristics of bioinformatics workflows were found mainly on the 0^{th} iteration (e.g., complex and repetitive tasks, quality checks, thresholds splitting of tasks, many files). Additional feedback lead to *tool* and *diagramSeparators* in the first iteration; while in the second iteration we added *standardReference* concept and the attributes *tool settings and parameters* for the meta-model extension, as well as the possibility to document *concurrent steps* in the WDST. Although these concepts arose specifically for bioinformatics workflows, of course they may be useful in other contexts. Three of these concepts (namely, *diagramSeparators*, *standardReference*, and *tool*) with its attributes, were not found in any related work, but were requested by the domain experts, leading us to believe they may be more specific to bioinformatics. Generally, any individual-driven workflow with many tools, scripts and file exchanges may require similar concepts.

RQ1.2: We employed the theories, Visual Alphabet and Physics of Notations [14] to visualize the concepts from RQ1.1. In the first and second iterations, the feedback received was compatible with these theories and did not result in any deletion, while in the last iteration, four concepts and notations were seen as unnecessary. We believe that the change of heart was due to the group discussion, resulting in the participants' confidence to reject concepts.

Moreover, the UML AD extension in this paper has a high graphical complexity, measured by the size of its visual vocabulary, containing 14 standards and nine extended notations, totalising 23 shapes (Table 1). Even though the complexity is high, the participants mentioned an average understandability of 4.3. Finally, some participants mentioned that the shapes were not intuitive when validating the concrete syntax. Therefore, we recommend that future use of our concrete syntax comes with textual labels for each shape and link.

Participants' feedback reveal a preference for their current unstructured (i.e., without a meta-model or set syntax) graphical representations rather than the developed notations, because using the former requires less knowledge about the modeling language and more about the context. Overall, we see a general reluctance to use a structured modeling language with a meta-model. However, we believe the drive towards open science will make such models increasingly necessary when boxes and arrows are too inexpressive and subject to interpretation.

RQ1.3: The WDST was envisioned for elicitation when it was created; however, during the first evaluation, the participants said that they would draw a diagram first and then fill the documentation. Therefore, we changed the WDST purpose from *workflow elicitation* to *documentation*. Even after this change, the participants preferred the diagrams over the WDST. Initially, we introduced a textual version of the workflow language with the idea that non-modelers may be more comfortable with the text. However, although bioinformaticians typically do not have training in modeling, they seem to prefer diagrams over text.

Overall, the WDST was a unanimously disliked template, with only negative average scales ranging from 1 to 2. Nonetheless, three important findings were made: i) the participants want an automatically generated documentation; ii) it must contain the tools settings and parameters; and iii) the amount of text and technicality should be as low as possible. We believe that an automatically generated documentation after drawing the workflow is the best solution.

4.1 Threats to Validity

Internal validity: The lack of bioinformaticians resulted in the availability of only seven participants, considered representative and having a mixed experience level. Some of the bioinformaticians participated in more than one round; thus, there is a gradual learning effect. However, we anticipate that the resulting language would be used more than once on a long-term basis; thus learning is a reasonable evaluation context. One of the drawbacks of group activities is the possibility for individuals to avoid taking part in the discussions and follow the crowd. To mitigate that, the seven participants were paired during the discussions to stimulate participation and prevent inhibition.

The researchers observed that the participants were avoiding answering the questions related to the WDST usage, addition, and removal of fields, by providing evasive and polite answers. As a mitigation, the validation question in the final iteration was performed entirely anonymously using Mentimeter. This approach revealed the participants' real thoughts about WDST.

The participants and interviewees were not native English speakers and did not share the same domain expertise. Additionally, the three involved facilities had a divergence of concepts. However, we adopted a simple language while interacting with the participants, created discussion sections, asked follow-up questions, and provided clarifications to mitigate any misunderstandings.

Reliability: To increase the reliability of our qualitative coding, one researcher created the code frame with its description and matching statements, while the other researcher independently checked reliability looking at the correspondence between the codes and the data [8]. We believe that other authors would create nearly the same concepts of this study but give them different names depending on their origin field and other factors. These additions were justified by the findings on 0^{th} iteration and the participants' validation.

External validity: We have used purposive sampling in this work. To address generalizability, three facilities took part during this study, and the participants worked with different workflows or different ways of designing workflows.

5 Related Work

Requirements Elicitation and Templates. In the requirements elicitation process, information is collected from stakeholders and end-users to understand system needs. In this case, we want to understand workflows and associated issues. General, requirements templates exist in the literature, e.g., the Volere template from Robertson and Robertson [18].

There are few approaches specifically for elicitation for bioinformatics. Work in [10] aimed to document workflow specifications for genomics data analysis. The workflow specifications consisted of the prescribed steps, until reaching a particular conclusion, including information about the specific tool versions with their parameter settings. However, the authors focused on using pre-built pipelines and standardized workflow definitions, where we focused on creating a language to facilitate standardized workflow documentation to provide an understandable and shareable view among collaborating bioinformaticians in projects.

Further work used semantic web standards to improve data workflow systems allowing bioinformaticians to publish and share their workflows via the cloud, providing an open collaboration between experts for workflow reproducibility, reusability, and data provenance [11]. Although the aims are similar, our approaches are different but potentially complementary.

UML Extensibility Mechanisms and Extensions. The creation of UML stereotype profiles allows UML meta-model extension and adaptation while keeping the existing UML syntax and semantics of the elements[16]. These stereotypes can have a different abstract syntax and extend either a meta-model class or another profile in a light-weight way, e.g., [12]. However, there is still no specific profile found for bioinformatics domain.

The literature covers several attempts to extend the UML AD meta-model for fields such as context-aware systems [1], production systems [3], project management [24], and business processes [22]. Although these extensions are not aimed for bioinformatics, some of these concepts and notations are useful and

align with the needs found in this work. Therefore, we have used this work as inspiration (see Table 1 for more detail).

UML has been used previously for bioinformatics workflows. For example, the authors in [19] evaluated UML use for specifying biological systems and processes, aimed for analysis, simulation, and prediction. However, this work does not focus specifically on the human-oriented workflow issues that we address.

6 Conclusion

The current state of bioinformatics workflow documentation is subjective and unstandardised. This paper presents a UML AD extension with its concrete syntax and a WDST as one of the first attempts to provide a language for a standard representation, where bioinformaticians validated the proposed concrete syntax as understandable and straightforward. According to the bioinformaticians, this extension would be used to document standard workflows, usually requested by stakeholders. The created WDST requires refinement and automation to be used for knowledge sharing and documentation by the bioinformaticians, as it was evaluated negatively. Much of the negative feedback we received was directed towards the tool (draw.io) and not the specifics of the language. We suggest further investigation, including the exploration of other modeling tools and frameworks (e.g. ADOxx or Eclipse Sirius).

We hope to validate the concepts with a broader bioinformatics community. Finally, future work should use our new language to assess and improve workflows, including making decision criteria clearer and adding more workflow automation when possible.

Acknowledgements. This work was supported by a Chalmers ICT Area of Advance SEED project and the Swedish Foundation for Strategic Research (RIF14–0081).

References

1. Al-alshuhai, A., Siewe, F.: An extension of uml activity diagram to model the behaviour of context-aware systems. In: 2015 IEEE Int. Conf. on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing. pp. 431–437. IEEE (2015)
2. Amstutz, P., Crusoe, M.R., Tijanić, N., Chapman, B., Chilton, J., Heuer, M., Kartashov, A., Leehr, D., Ménager, H., Nedeljkovich, M., et al.: Common workflow language, v1. 0 (2016)
3. Bastos, R.M., Ruiz, D.D.A.: Extending uml activity diagram for workflow modeling in production systems. In: Proc. of the 35th Annual Hawaii Int. Conf. on System Sciences. pp. 3786–3795. IEEE (2002)
4. Brooke, J.: Sus: a retrospective. Journal of usability studies **8**(2), 29–40 (2013)
5. Fernando, T., Gureev, N., Matskin, M., Zwick, M., Natschläger, T.: Workflowdsl: Scalable workflow execution with provenance for data analysis applications. In: 2018 IEEE 42nd Annual Computer Software and Applications Conf. (COMPSAC). vol. 1, pp. 774–779. IEEE (2018)

6. Figueroa, L.H.B.d., Salman, R., Horkoff, J., Chauhan, S., Davila, M., de Oliveira Neto, F., Schliep, A.: A Modeling and Elicitation Approach for Bioinformatics Workflows: Supporting Material. Online (2019), http://www.cse.chalmers.se/~jenho/BioinformaticsWorkflows/
7. Güss, C.D.: What is going through your mind? thinking aloud as a method in cross-cultural psychology. Frontiers in psychology **9**, 1292 (2018)
8. Harper, D., Thompson, A.R.: Qualitative research methods in mental health and psychotherapy: A guide for students and practitioners. John Wiley & Sons (2011)
9. Hevner, A.R.: A three cycle view of design science research. Scandinavian journal of information systems **19**(2), 4 (2007)
10. Kanwal, S., Lonie, A., Sinnott, R.O.: Digital reproducibility requirements of computational genomic workflows (2017)
11. Karim, M.R., Michel, A., Zappa, A., Baranov, P., Sahay, R., Rebholz-Schuhmann, D.: Improving data workflow systems with cloud services and use of open data for bioinformatics research. Briefings in bioinformatics **19**(5), 1035–1050 (2017)
12. Korherr, B., List, B.: Extending the uml 2 activity diagram with business process goals and performance measures and the mapping to bpel (2006)
13. Krishna, R., Elisseev, V., Antao, S.: Baas-bioinformatics as a service. In: European Conf. on Parallel Processing. pp. 601–612. Springer (2018)
14. Moody, D.: The "physics" of notations: toward a scientific basis for constructing visual notations in software engineering. IEEE Transactions on software engineering **35**(6), 756–779 (2009)
15. Moody, D.L., Heymans, P., Matulevicius, R.: Improving the effectiveness of visual representations in requirements engineering: An evaluation of i* visual syntax. In: 2009 17th IEEE Int. RE Conf. pp. 171–180. IEEE (2009)
16. OMG: OMG Unified Modeling Language (OMG UML), Superstructure, Version 2.4.1 (August 2011), http://www.omg.org/spec/UML/2.4.1
17. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A design science research methodology for information systems research. Journal of management information systems **24**(3), 45–77 (2007)
18. Robertson, S., Robertson, J.: Mastering the requirements process: Getting requirements right. Addison-wesley (2012)
19. Roux-Rouquié, M., Caritey, N., Gaubert, L., Rosenthal-Sabroux, C.: Using the unified modelling language (uml) to guide the systemic description of biological processes and systems. Biosystems **75**(1-3), 3–14 (2004)
20. Smith, J.A.: Qualitative psychology: A practical guide to research methods. Sage (2015)
21. Spyrou, S., Bamidis, P., Pappas, K., Maglaveras, N.: Extending uml activity diagrams for workflow modelling with clinical documents in regional health information systems. In: Connecting Medical Informatics and Bioinformatics: Proc. of the 19th Medical Informatics Europe Conf. (MIE2005). pp. 1160–1165 (2005)
22. Stefanov, V., List, B., Korherr, B.: Extending uml 2 activity diagrams with business intelligence objects. In: Int. Conf. on Data Warehousing and Knowledge Discovery. pp. 53–63. Springer (2005)
23. Störrle, H.: Semantics of structured nodes in uml 2.0 activities. In: 2nd Nordic Workshop on UML. pp. 19–32 (2004)
24. Syriani, E., Ergin, H.: Operational semantics of uml activity diagram: an application in project management. In: 2012 Second IEEE Int. Workshop on Model-Driven Requirements Engineering (MoDRE). pp. 1–8. IEEE (2012)
25. Taherdoost, H.: Sampling methods in research methodology; how to choose a sampling technique for research (2016)