applied sciences

MDPI

Article

Process Model Metrics for Quality Assessment of Computer-Interpretable Guidelines in PROforma

Joaquín Torres-Sospedra [1,*,†], Begoña Martínez-Salvador [2,*,†], Cristina Campos Sancho [3] and Mar Marcos [2]

[1] Institute of New Imaging Technologies, Universitat Jaume I, 12071 Castellón, Spain
[2] Department of Computer Engineering and Science, Universitat Jaume I, 12071 Castellón, Spain; mar.marcos@uji.es
[3] Department of Languages and Computer Systems, Universitat Jaume I, 12071 Castellón, Spain; camposc@uji.es
[*] Correspondence: jtorres@uji.es (J.T.-S.); begona.martinez@uji.es (B.M.-S.)
[†] These authors contributed equally to this work.

Abstract: Background: Clinical Practice Guidelines (CPGs) include recommendations to optimize patient care and thus have the potential to improve the quality and outcomes of healthcare. To achieve this, CPG recommendations are usually formalized in terms of Computer-Interpretable Guideline (CIG) languages. However, a clear understanding of CIG models may prove complicated, due to the inherent complexity of CPGs and the specificities of CIG languages. Drawing a parallel with the Business Process Management (BPM) and the Software Engineering fields, understandability and modifiability of CIG models can be regarded as primary quality attributes, in order to facilitate their validation, as well as their adaptation to accommodate evolving clinical evidence, by modelers (typically teams made up of clinical and IT experts). This constitutes a novel approach in this area of CIG development, where understandability and modifiability aspects have not been considered to date. Objective: In this paper, we define a comprehensive set of process model metrics for CIGs described in the PROforma CIG language, with the main objective of providing tools for quality assessment of CIG models in this language. Methods: To this end, we first reinterpret a set of metrics from the BPM field in terms of PROforma and then we define new metrics to capture the singularities of PROforma models. Additionally, we report on a set of experiments to assess the relationship between the structural and logical properties of CIG models, as measured by the proposed metrics, and their understandability and modifiability from the point of view of modelers, both clinicians and IT staff. For the analysis of the experiment results, we perform statistical analysis based on a generalized linear mixed model with binary logistic regression. Results: Our contribution includes the definition of a comprehensive set of metrics that allow measuring model quality aspects of PROforma CIG models, the implementation of tools and algorithms to assess the metrics for PROforma models, and the empirical validation of the proposed metrics as quality indicators. Conclusions: In light of the results, we conclude that the proposed metrics can be of great value, as they capture the PROforma-specific features in addition to those inspired by the general-purpose BPM metrics in the literature. In particular, the newly defined metrics for PROforma prevail as statistically significant when the whole CIG model is considered, which means that they better characterize its complexity. Consequently, the proposed metrics can be used as quality indicators of the understandability, and thereby maintainability, of PROforma CIGs.

Keywords: software models; process models; computer-interpretable guidelines; metrics; process model quality

check for updates

Citation: Torres-Sospedra, J.; Martínez-Salvador, B.; Campos Sancho, C.; Marcos, M. Process Model Metrics for Quality Assessment of Computer-Interpretable Guidelines in PROforma. *Appl. Sci.* **2021**, *11*, 2922. https://doi.org/10.3390/app11072922

Academic Editor: Ricardo Colomo-Palacios

Received: 19 February 2021
Accepted: 19 March 2021
Published: 25 March 2021

1. Introduction

1.1. Background

According to the most recent definition, trustworthy Clinical Practice Guidelines (CPGs) are "statements that include recommendations intended to optimize patient care that are informed by a systematic review of evidence and an assessment of the benefits and harms of alternative care options” [1]. Research has demonstrated that CPGs have the potential to facilitate the translation of clinical research results into practice and to improve the quality and outcomes of healthcare. To achieve these benefits, CPG recommendations should be made available to clinicians where and when they are needed [2]. Although this can be done using CPGs in their original text form, there is a consensus that the most effective way is by converting them into a computer-interpretable format [3]. Thus, Computer-Interpretable Guidelines (CIGs) can be defined as formalized versions of CPG contents intended to be executed as part of decision-support systems.

It is well established that CPGs are difficult to comprehend and formalize [4]. CPG texts can be semantically very complex, with rich content in knowledge of very varied type (procedures, decision criteria, abstract concepts, etc.). In line with this, CIG representation languages provide a wide range of modeling constructs tailored to these knowledge types. This makes CIG languages poorly accessible and understandable for clinicians, in general. As a result of these factors, the formalization of CPGs is usually carried out by joint teams made up of clinical and IT experts. On the one hand, specialized clinical knowledge is required for an adequate understanding of most of CPG contents. On the other hand, knowledge engineering skills are necessary to analyse and identify the CPG contents amenable to formalization, as well as to describe them in terms of the CIG language of choice. Even with such a collaborative approach, a clear understanding of the final CIG model may prove complicated, due to the inherent complexity of the CPG and to the specificities of the CIG language, among other factors.

Against this background, the assessment of the understandability and modifiability of CIG models gains special relevance. An important reference topic is comprehension of business process models, given the similarities between the specific part of CIG languages dedicated to procedural knowledge and Business Process Management (BPM) notations. This parallelism has been recognized and exploited for some time in several works (see, e.g., in [5–7]). Understandability and modifiability of CIG models can be regarded as primary quality attributes, in order to facilitate the validation of the model as well as its adaptation to accommodate evolving clinical evidence.

In the BPM literature, several works deal with different aspects with an influence on the understandability of process models. One of them is the heuristics of the so-called Seven Process Modeling Guidelines [8]. Inspired by the long tradition in Software, other works use metrics to capture the structural and logical complexity of process models and analyze how the metric values relate to their understandability and modifiability [9]. In this line, in a previous work [10] we analyzed the metrics proposed in the BPM area by Mendling [11] and reinterpreted them in terms of a specific CIG language, PROforma [12].

PROforma provides a principled approach to modeling the logical and procedural aspects of clinical decision making. It is a representative language of the so-called Task-Network Model (TNM) approach [2], which consists in describing guidelines in terms of a hierarchical decomposition of networks of component tasks. A highly distinctive feature of CIG languages lies in the model used to represent decision-making policies. In this regard, PROforma is characterized by the use of a decision model based on argumentation logic. PROforma stands out for having multiple positive assessments in clinical settings [13]. Among these, quantitative trials have been carried out which have demonstrated positive effects of the PROforma systems in healthcare outcomes [14].

In this paper, we take a step further and define a comprehensive set of process model metrics for CIGs described in the PROforma language, with the main objective of providing tools for quality assessment of CIG models. Note that we focus on process model metrics for PROforma CIGs, rather than on the modeling thereof. The proposed metrics not only encompass process model aspects considered in the BPM field, but also others that take into account the specificities of PROforma models, including the richness of the description of decision logics and the intensive use of hierarchical decomposition of tasks. In addition, we report on a set of experiments to assess the relationship between the structural and logical properties of CIG models, as measured by the proposed metrics, and their understandability and modifiability. Importantly, we are concerned with the understandability and modifiability of PROforma CIGs from the point of view of modelers in general, i.e., both clinicians and IT experts with an adequate training in the language.

1.2. Related Work

CIG quality is a crucial aspect that can be considered from different perspectives. In the medical field, there is a concern about how to evaluate the quality of CPG documents as well as how these documents can meet certain quality standards. A number of tools have been developed for this purpose, e.g., instruments to assess the quality of both the CPG development process and the CPGs themselves [15], and approaches for rating the strength of CPG recommendations and the quality of the evidence supporting them [16]. We shall not dwell further on the quality of CPGs, as it is outside of the scope of our work. Focusing on CIG quality aspects, it is imperative to consider the topic of Software Quality Assurance (SQA), in addition to the above-mentioned topic of business process model comprehension.

SQA is concerned with ensuring and demonstrating that a software system satisfies the needs and requirements of the customer, and that it will continue to satisfy them in the future [17]. The latter involves aspects such as the ease to fix the software and to adapt it to new requirements. Starting in the late seventies, several frameworks have been proposed defining the fundamental characteristics to evaluate the quality of software [18]. Modifiability (or flexibility), defined as the degree to which the code facilitates the incorporation of changes, appears in McCall's quality model already in 1977. Boehm's model improves McCall's one adding characteristics such as understandability (or recognizability), defined as the degree to which the software allows users to recognize whether it is appropriate for their needs. Stressing the importance of customer satisfaction, the newest ISO 25010 quality model [19] defines a *quality in use* model separate from the *product quality* model [20]. The latter encompasses eight software quality characteristics and 31 sub-characteristics, including the modifiability and understandability.

Verification & Validation (V&V) activities are central to SQA [20]. Verification aims to determine whether the software product under construction matches its specification (i.e., building the product right), while the goal of validation is to determine whether the software satisfies the needs of the customer (i.e., building the right product). Verification is usually performed by examining descriptions of the software (e.g., requirements, specifications, and code), while validation relies on testing the software in execution. Testing should be performed throughout the software development process, and not only at the end of it [20]. If this is done, testing can uncover problems in specifications or errors prior to the delivery of the final software. However, it may be difficult to ensure that a software is correct using testing, e.g., due to the huge number of test cases required, or due to ill-defined requirements. In such cases, verification based on reviews and inspections involving the customer play a fundamental role. Formal methods can also be used to verify that a software will operate correctly, by means of mathematical tools to model a system and check that this model fulfills a series of desirable properties. Additionally, there is a long tradition of using software metrics to evaluate the quality of design, with the aim of detecting certain maintainability issues.

With regard to the latter topic, several works propose metrics to evaluate different quality aspects regarding the characteristics of software and business process models. Canfora et al. [21] propose a set of metrics for software process models and describe a family of experiments conducted to validate if these metrics are suitable as model quality indicators. In particular, the authors focus on the three sub-characteristics of maintainability: analyzability, understandability, and modifiability. The metrics are based on the main elements represented in a software process model, and include, e.g., *number of activities* or *ratio of work products and activities*. The conclusions of the experiments indicate that these metrics are good maintainability indicators. In the BPM field, several works related to the quality of business process models have been published recently.

Mendling [11] analyzed different metrics from the areas of network analysis, Software Engineering, and BPM, and proposed a set of 15 metrics that deal with different aspects of the structure and space state of the process model. Sánchez-González et al. [22] adapted these metrics to the OMG standard Business Process Modelling and Notation (BPMN) and performed a series of experiments with exercises aimed to analyze the correlation between correct answers and metric values. Recently, Hasić and Vanthienen have done a similar work by defining a set of metrics for the new OMG standard Decision Model and Notation (DMN), intended to complement BPMN with the modeling of decisions [23]. These works are in line with the general viewpoint in the Software Engineering field, where there is a range of frameworks addressing the quality of conceptual models [24]. Moody argues that one of the major obstacles hindering the use of this kind of metrics is in many cases the lack of empirical evidence to support them, despite the fact that it is needed to promote their acceptance in practice [24].

In the case of CIGs, we can say that clinical experts are the customers and CPG texts constitute the requirements [2]. Validation and verification of CIG models are usually performed by clinical experts assisted by knowledge engineers. Validation in general includes testing the CIG with different patient data values (simulated or real) to check whether the resulting recommendations are as anticipated by the CPG (and the clinical experts). Additionally, clinical experts can inspect the CIG trying to detect flaws in the implemented logic. This inspection extends to the different CPG properties and indicators that have been defined and formalized, if any. With respect to verification, there are two main lines of work: proving that the CIG is consistent and free of anomalies, and proving that it satisfies a set of desirable properties [2]. The latter has been done using formal verification methods, including theorem proving [25] and model checking [26]. Our work is placed in the context of CIG quality assurance and complements previous work in this area. With respect to the BPM area, our work has roots in the previously defined BPMN metrics and is an effort parallel to the definition of metrics for the DMN standard. Even when a CIG is internally consistent, free of anomalies, and satisfies a set of predefined properties, there exist quality aspects related to the process models that might be interesting to quantify to determine if those models are easily understandable and modifiable.

2. Materials and Methods

2.1. The PROforma Language

PROforma [12] is a formal knowledge representation language tailored to capture clinical knowledge. It is a well-established language which has been (and is still being) successfully used for the deployment and execution of clinical guidelines models [13,14,27]. PROforma is supported by several software tools, including an execution engine and different editing environments. It was designed with the aim to integrate into clinical process descriptions an explicit model of decision-making, with an expressivity and level of detail that BPM languages lack [14]. Recent work by the OMG aims to complement the BPMN notation providing support for the description of cases (CMMN) and business decisions and rules (DMN) [28,29]. However, DMN does not allow for the possibility of combining arguments for and against a hypothesis (or candidate) to derive a decision, as PROforma does (see below). This feature is particularly suitable for clinical decision-making, which makes PROforma the preferable option over BPM languages.

In PROforma, a guideline is modeled as a plan made up of one or more *tasks*. There are four types of tasks: *actions*, *enquiries*, *decisions*, and *plans*. An action corresponds to an activity (e.g., a clinical procedure) to be performed by an external agent. An inquiry is a task that acquires information, i.e., the value of one or more data items or *sources*, from the external environment (e.g., clinician and databases). A decision is a task that represents a choice among different *candidates* (e.g., low or high risk level). Finally, a plan is a container that can be used to group together a set of other tasks. As a plan may contain in turn other nested plans, PROforma allows the definition of hierarchical task networks. The tasks within a plan are usually ordered via *scheduling constraints* and/or different kinds of *task conditions*. If none are given, a parallel execution of tasks is assumed.

In the PROforma graphical notation, CIGs are depicted as directed graphs in which nodes represent tasks and arcs represent scheduling constraints. In this notation, the shape of the nodes indicates the task type: squares are used for actions, circles for decisions, diamonds for inquiries, and round-edged rectangles for plans. In the case of scheduling constraints, the arc indicates that the task at the head of the arc cannot start until the task at the tail of the arc (antecedent task) has completed. An example of PROforma graph can be found in Figure 1.

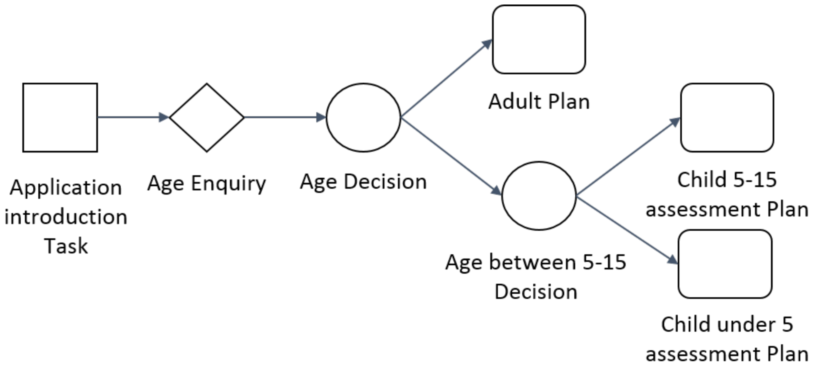

Application introduction Task
Age Enquiry
Age Decision
Adult Plan
Age between 5-15 Decision
Child 5-15 assessment Plan
Child under 5 assessment Plan

Figure 1. Top-level plan of Asthma PROforma CIG. Squares are used for actions, circles for decisions, diamonds for inquiries, and round-edged rectangles for plans. This CIG is represented as a directed graph with nodes of different shapes: squares for actions, circles for decisions, diamonds for inquiries, and round-edged rectangles for plans.

More specifically, the execution of a PROforma CIG evolves as follows. A task can only be considered for activation when all its scheduling constraints have been met, i.e., when all its antecedent tasks have been either completed or discarded. In that case, the task will be activated if at least one of the antecedent tasks has completed; otherwise, it will be discarded. Tasks may also have different types of conditions imposing additional constraints to be met before activation, including *preconditions* and *wait conditions*. Both are truth-valued expressions that are checked when the scheduling constraints are met. In the case of preconditions, the task will be activated if the precondition holds; otherwise, it will be discarded and will not be considered again. In the case of wait conditions, the task will remain dormant until the condition is met, with the possibility of a subsequent reactivation if the condition is met again. For more details on PROforma, see the OpenClinical.net (accessed on 29 April 2020) resources [30].

The specification of PROforma decisions requires additional attributes, apart from the associated candidates. Notably, each candidate can have one or more *arguments*, which are truth-valued expressions that determine the choice of that candidate. These expressions usually describe the arguments for (in favor) or against the candidate. Additionally, each candidate has a *recommendation rule*, which is an expression that is used to calculate the support for the candidate considering all its arguments. Finally, the choice mode, single or multiple, determines how many candidates can be recommended by the decision. As an illustration, Figure 2 shows the PROforma code of the task *Age between 5–15 Decision* from Figure 1. This decision has two candidates: *age_between_5_and_15* and *age_not_between_5_and_15*. Each candidate has one argument with a condition to confirm the corresponding candidate and a recommendation rule that establishes that this candidate will be selected when it is supported by at least one argument. This decision task has both a scheduling constraint and a precondition. This implies that it will be activated only when the antecedent task has completed, whenever the expression in the precondition holds.

```
decision  :: 'dec_age_5_15' ;
caption :: 'Age between 5-15 Decision' ;
precondition :: result_of(dec_age_15) = age_not_greater_than_15;
candidate :: 'age_between_5_and_15' ;
argument :: confirming,(Age >= 5 and Age <= 15)               attributes
argument_name :: ' ( age >= 5 AND age =< 15 ) ' ;
end attributes
;
recommendation :: Netsupport( dec_age_5_15, age_between_5_and_15 ) >= 1 ;
candidate :: 'age_not_between_5_and_15' ;
argument :: confirming,(Age < 5)                 attributes
argument_name :: ' ( age < 5 ) ' ;
end attributes
;
recommendation :: Netsupport( dec_age_5_15, age_not_between_5_and_15 ) >= 1;
end decision.
```

Figure 2. PROforma task "age between 5–15 decision".

2.2. Metrics for the Evaluation of PROforma Models

In this section, we describe the metrics we propose to assess quality aspects of CIGs modeled in PROforma related to their understandability and modifiability. Note that a high (respectively, low) metric value does not necessarily imply that the model is low (high) quality. Instead, given two equivalent models, the metric value could identify which model is the least understandable and therefore error-prone. Our starting point is the set of 15 metrics proposed by Mendling [11] dealing with different aspects of the structure and the space state of a process model: size, coupling, cohesion, complexity, and modularity. The size aspect measures how big the process is or how many elements it has. Coupling deals with the number of interconnections among the different modules of the process model, the density of these interconnections, and how complex they are. Cohesion measures the relationship among the elements inside a module. Complexity deals with the simplicity of the process models, usually related to the number of control flows and the number of modules of the process. Finally, modularity measures to what extent a process can be separated into different parts or modules. These 15 metrics have been adapted to the BPMN language and validated by Sanchez-Gonzalez [22]. In order to reinterpret them in terms of PROforma, first we need to consider the main differences between both formalisms. Moreover, it is crucial to consider the specific types of knowledge typically contained in CPGs. Taking into account both aspects, we have reinterpreted the original metrics by Mendling, and further we have defined new PROforma-specific ones, as we explain in this section.

Mendling et al. [11] considered the process model as a graph $G = (N, A)$, where N is the set of nodes and A is the set of arcs connecting those nodes, $A \subseteq N \times N$. We have also considered a PROforma model as a graph, where the tasks correspond to nodes and the scheduling constraints correspond to arcs. Clinical processes included in guidelines do not usually contain unstructured loops. Then, we have ruled out the possibility of arbitrary cycles in the process model graph. As a matter of fact, the PROforma editor warns about graph cycles.

In BPMN there are different types of nodes: tasks, split connectors, and join connectors. In contrast, PROforma does not have connector nodes properly speaking. Instead, we have considered any task (node) with more than one incoming or more than one outgoing scheduling constraint (arc) as a connector node. Accordingly, if the number of outgoing arcs is greater than one, we talk about split connectors, and, if the number of incoming arcs is greater than one, they are join connectors. In BPMN, parallel, exclusive or inclusive split connectors are explicitly distinguished, regarding the behavior of the split node on the number of available paths taken (all, one or variable, respectively). However, in PROforma, we do not make any difference among split connectors regarding their behavior as it is not possible to determine it from the node. For example, if the connector node is a PROforma decision task, its actual behavior (parallel, exclusive or inclusive) is determined not only by the result of the decision but also by the preconditions of the subsequent tasks.

One of the distinctive features of PROforma is decision tasks. According to our experience, decisions tasks can be very complex, and therefore their use can have a negative impact in the understandability of the process model. For this reason, we have considered to define new metrics to study their impact in the quality of the model.

PROforma is a hierarchical task network language, and therefore plans are also a key feature of the models. A plan is a task that acts as a container for other tasks, which are grouped together to achieve a particular goal. Although BPMN allows for subprocesses, the set of metrics by Mendling [11] does not consider this feature. Notice that the use of plans is related to modularity, which is a key principle to deal with complex processes and usually it facilitates the understandability of complex models [31]. However, the decomposition of a process into plans depends on the knowledge engineer criteria, and as Reijers et al. [31] argued, there is no objective benchmark on how to use it. Moreover, it seems that when the process logic is fragmented across several sub-plans, decomposition might become a drawback instead of a benefit. For that reason, we have defined new metrics to study their effect in the understandability and modifiability aspects of the model.

Therefore, considering the differences between PROforma models and BPMN models described in the preceding paragraphs, we have redefined the initial set of metrics as follows:

- We have not considered the cyclicity metric, since we have ruled out the possibility of arbitrary cycles (see above).
- We have referred to connectors instead of to gateways in the following metrics: average connector degree (Section 2.2.2), maximum connector degree (Section 2.2.2), and connector mismatch (Section 2.2.4). In the calculation of the token split metric (Section 2.2.5), we have also simplified the join connectors, having a single type instead of distinguishing between AND, OR, and XOR joins.
- We have not considered the behavior of the different split nodes. For that reason, we have ruled out the gateway heterogeneity metric that measures the type entropy of the gateways. Besides, the metrics connector mismatch and control flow complexity (Section 2.2.4) have been redefined considering that all the split connector nodes have the same behavior.

Moreover, we have also defined new metrics in order to consider the specific characteristics of PROforma. First, in order to capture the impact of decisions in the complexity of the model, we have formulated the following new metrics: number of decisions, density of decisions, and number of preconditions.

Second, the use of plans in the clinical processes modeled with PROforma has several implications regarding the metrics and also brings about the definition of new metrics:

- Whenever there is more than one start task (i.e., tasks without any incoming scheduling constraint) and/or more than one end task (i.e., tasks without any outgoing scheduling constraint) in a plan, we have considered an implicit parallel split and/or join within the plan. Accordingly, dummy components (tasks and scheduling constraints) have been incorporated to account for these implicit splits and/or joints.
- We have defined some metrics in order to determine the effect of plans and their size in the quality of the model: number of plans, density of plans, average size of a plan, plans with a single task, and plans with a size above the average.
- For each metric, we have defined an aggregation method of the values obtained for every graph that is part of the model (see Section 2.2.6). These metrics provide a more comprehensive characterization of the entire model.

The complete collection of metrics for PROforma models is described below, providing a formula for those metrics that do not correspond to a single observable value. We have subdivided the set of metrics into the following six categories: size, density, partitionability, connector interplay, concurrency, and aggregation.

The elements within a PROforma plan have been represented as a directed graph (G) where tasks are nodes and scheduling constraints are arcs. We have used the following notation:

- N: set of nodes—actions, inquiries, decisions, and plans—(see Figure 1 for examples of each one of these elements),
- A: set of arcs, i.e., scheduling constraints between a pair of tasks,
- P: set of plans, i.e., nodes that correspond to PROforma plans,
- CT: set of all connector tasks, i.e., nodes with more than one incoming arc and nodes with more than one outgoing arc,
- NCT: set of all non-connector tasks ($NCT = N - CT$).
- SCT: set of split connector tasks.
- JCT: set of join connector tasks ($CT = SCT + JCT$).

In the formulas that follow, unless otherwise indicated, N, A, P, CT, NCT, SCT, and JCT refer to the corresponding sets of a single graph (plan) and not to the entire model.

2.2.1. Size Metrics

Size is usually an important factor in the understandability of process models. Usually size is related to the number of nodes N. Furthermore, we have considered the diameter of the process graph, the number of arcs or scheduling constraints, and the new metric number of plans.

- *Size*, S_N: the number of tasks (nodes) in the graph.

$$S_N(G) = |N|$$

- *Arcs*, S_A: the number of scheduling constraints in the graph

$$S_A(G) = |A|$$

- *Diameter*, *diam*: length of the longest path from a start task to an end task in the graph.
- *Number of plans*, S_P: the number of tasks in the graph that correspond to plans.

$$S_P(G) = |P|$$

2.2.2. Density Metrics

We have used density as a generic term to refer to any metric that relates the quantity of two elements of the graph. In this category fall the metrics density, coefficient of connectivity, average connector degree, and maximum connector degree. Moreover, the newly defined metrics related to plans and decisions also fall in this category: density of plans, percentage of single-node plans, average size of a plan, percentage of plans whose size is above average size, and decision density.

- *Density*, Δ: it measures how far or close is the number of arcs to the maximal number of arcs. It is computed as the ratio of scheduling constraints to the maximum number of scheduling constraints.

$$\Delta(G) = \frac{|A|}{|N| \cdot (|N| - 1)}$$

- *Coefficient of connectivity*, CNC: it is a measure of how dense is the graph regarding the number of connections. It is computed as the ratio of scheduling constraints to tasks.

$$CNC(G) = \frac{|A|}{|N|}$$

- *Average connector degree*, $\overline{d_C}$: it is a measure of the number of nodes a connector is in average connected to. It is computed as the average number of scheduling constraints of connector tasks, where $d(c)$ is the number of scheduling constraints of the connector task c. The metric considers both the incoming and the outgoing scheduling constraints.

$$\overline{d_C}(G) = \frac{1}{|CT|} \sum_{c \in CT} d(c)$$

- *Maximum connector degree*, $\widehat{d_C}$: it is the maximum number of nodes a connector is connected to. It is computed as the maximum number of scheduling constraints of connector tasks. As in the previous metric, all scheduling constraints are included.

$$\widehat{d_C}(G) = max\{d(c)/c \in CT\}$$

- *Density of plans*, $\overline{p}$: it measures the level of clustering in a process model. It quantifies how many among the total number of tasks in the graph are plans, and it is computed as the ratio of the number of plans to the total number of tasks of the graph.

$$\overline{p}(G) = \frac{S_P(G)}{S_N(G)}$$

- *Percentage of single-node plans*, $p1\%$: it is a measure of how fragmented is the model respect to single-node plans. Single-node plans capture excessive fragmentation. This metric is computed as the ratio of plans that contain a single node to the total number of plans.

$$p1\% = \frac{|\{p \in P/S_N(G_p) = 1\}|}{|P|}$$

  where G_p is the graph representing the content of plan p. In this metric, P denotes the set of plans of the entire model (not a single graph).
- *Average size of a plan*, $\bar{t}$: t is a measure of how dense is a plan. That is, how many tasks plans have on average. In order to compute it, we have consider the size of all the plans of the model (P).

$$\bar{t} = \frac{1}{|P|} \sum_{i=1}^{|P|} |S_N(G_i)|$$

- *Percentage of plans whose size is above average size*, $p50\%$: it is measure of how homogeneous plans are in size. It is computed as the number of plans whose size is above average (see previous metric) to the total number of plans.

$$p50\% = \frac{|\{p \in P/|S_N(G_p)| > \bar{t}\}|}{|P|}$$

- *Decision density*, δ_D: it is a measure of how dense is the graph respect to this specific PROforma element. It is computed as the ratio of the number of nodes that correspond to decisions to the total number of tasks in the graph. The number of decisions, S_D, is a new metric described in Section 2.2.4.

$$\delta_D(G) = S_D(G)/S_N(G)$$

Note that the metrics related to the density of the plans (average size, percentage of single-node plans, and percentage of plans of size above average) are not defined for a single plan (graph) but for the full model.

2.2.3. Partitionability Metrics

Partitionability is used for referring to the relationship of subcomponents to the overall model. Within this category are the metrics separability, sequentiality, structuredness, depth, and the new metric model depth.

- *Separability*, Π: it tries to capture how far certain parts of the model can be considered in isolation. An increase in the value of this metric might imply a simpler model. It is computed as the ratio of cut vertices to tasks. A cut vertex (or articulation point) is a node whose deletion separates the graph into several components.

$$\Pi(G) = \frac{|\{n \in N/n \text{ is a cut vertex}\}|}{|N| - 2}$$

- *Sequentiality*, Ξ: it measures how sequential is a plan. This metric relates to the fact that sequences of nodes are the most simple components in a graph. It is calculated as the ratio of the maximum possible number of scheduling constraints between non-connector tasks to the total number of scheduling constraints.

$$\Xi(G) = \frac{|A \cap (NCT \times NCT)|}{|A|}$$

- *Structuredness*, ϕ: it measures how far a process model is made of nesting blocks of matching join and split connectors. For this metric, it is necessary to obtain the reduced process graph applying the graph reduction rules defined by Mendling [11]. Structuredness is computed as one minus the number of tasks in the reduced process graph, $|N'|$, divided by the number of tasks in the original process graph. The structuredness value for a structured graph is 1.

$$\phi = 1 - \frac{|N'|}{|N|}$$

- *Depth*, Λ: it is related to the maximum nesting of structured blocks in a graph. It is computed as the maximum depth of all nodes, where the depth of a node $\lambda(n)$ is calculated as the minimum of the in-depth and out-depth of the node. The in-depth $\lambda_{in}(n)$ refers to the maximum number of split connectors that must be traversed in a path reaching the node from the start node, minus the number of join connectors in the same path. The out-depth $\lambda_{out}(c)$ is defined analogously with respect to the end node.

$$\Lambda(G) = max\{\lambda(n)/n \in N\}$$

- *Model depth*, Υ: it computes the maximum nesting of a task in the hierarchy of plans. Starting at the top-level plan, where it would be initialized to 1, each time the process logic traverses a plan, it would be increased by one. Therefore, it can be defined as the maximum number of plans that it is necessary to descend to reach a task. We define the model depth of a task *t* recursively as follows (notice that in PROforma, plans are a type of tasks).

$$modelDepth(t) = 1,\ t \in top\_level\ plan$$

$$\begin{aligned} modelDepth(t) =& 1 + modelDepth(p), \\ & t \notin top\_level\ plan,\ t \in p,\ p \in P \end{aligned}$$

$$\Upsilon = maximum(modelDepth(t))\ \forall t$$

Note that this latter metric is different from the depth metric that considers the nesting of a task in a graph with respect to the split/join connections traversed. In contrast, model depth measures the nesting of a task considering the hierarchy of graphs. Although it is possible to compute the model depth of any plan, we have only considered the model depth metric of the top-level plan (full model).

2.2.4. Connector Interplay Metrics

This section presents the metrics related to connectors and their interplay, in particular, connector mismatch and control flow complexity metrics. As the behavior of some connectors in PROforma depends on the result of the decisions and on the preconditions of the tasks, we have included two new metrics in this category: number of decisions and number of preconditions.

- *Connector mismatch*, MM: this metric relates to the structuredness of the model, as this property implies that each split connector matches a corresponding join connector. The metric counts the number of mismatches of connector tasks, i.e., number of split connector tasks that do not have a corresponding join connector task. Since we do not have different split/join connectors, it is calculated as the difference between the sizes of both sets:
$$MM = ||SCT| - |JCT||$$
- *Control flow complexity*, CFC: it tries to measure how difficult is to consider all potential states after a split connector. It is computed as the sum of all split connectors tasks (SCT) weighted by the potential combinations of states after the split, i.e., $2^{d_{out}(c)} - 1$ where $d_{out}(c)$ is the number of outgoing scheduling constraints of the connector task c. Notice that in our models all connectors are considered *or*-connectors, the worst case scenario for a split connector.
$$CFC(G) = \sum_{c \in SCT} (2^{d_{out}(c)} - 1)$$
- *Number of decisions* S_D: In some cases, the behavior of some connectors depends on the result of decisions. This metric calculates the number of nodes of the graph that correspond to PROforma decision tasks.
- *Number of preconditions*, $S_{Precond}$: Related with the previous metric, the complexity of control flows is increased if they have preconditions to be evaluated. This metric counts the number of preconditions in the graph.

2.2.5. Concurrency Metrics

It is necessary to keep track of how many concurrent paths are in the graph in order to synchronize them. Split connectors tasks could potentially introduce new threads of control. This is measured by the token split metric. Concurrent paths introduced from the beginning (not by split connectors tasks) are not considered.

- *Token split*, TS: sum of output degrees minus 1 of all split connector tasks (SCT).

$$TS(G) = \sum_{c \in SCT} (d_{out}(c) - 1)$$

2.2.6. Aggregation Metrics

PROforma is a hierarchical task network language, therefore a CIG is represented in PROforma as a hierarchy of plans. Accordingly, for each metric we have defined an aggregation to better characterize the full model, except in the case of metrics which already consider all the graphs of the model (i.e., model depth and the metrics related to the size of plans). Hereinafter, these metrics will be referred as *full-model metrics*, while the ones computed for a single graph will be referred as *single-graph metrics*.

We have used different aggregation formula depending on the metric as shown in Table 1. For those metrics that count a particular element of the graph or of the PROforma language, namely, nodes, arcs, plans, decisions, and preconditions, the aggregation metric has been defined as the sum of the values. In the case of the diameter metric within the size category, the aggregation metric has been calculated as weighted average with respect to the number of nodes of each graph.

Table 1. Aggregation formula for the different graph metrics.

Metric	Aggregation Calculated as
Size S_N	Sum of values
Number of arcs S_A	Sum of values
Diameter $diam$	Weighted average of values
Number of plans S_P	Sum of values
Density Δ	Weighted average of values
Coefficient of connectivity CNC	Average of values
Average connector degree $\overline{d_C}$	Weighted average of values
Maximum connector degree $\widehat{d(c)}$	Maximum of values
Density of plans $\overline{p}$	Average of values
Decision density δ_D	Average of values
Separability Π	Weighted average of values
Sequentiality Ξ	Weighted average of values
Structuredness ϕ	Weighted average of values
Depth Λ	Weighted average of values
Connector mismatch MM	Weighted average of values
Control flow complexity CFC	Weighted average of values
Number of decisions S_D	Sum of values
Number of preconditions $S_{Precond}$	Sum of values
Token split TS	Weighted average of values

For some of the metrics that refer to densities, their aggregation has been defined as the average of the values. This is the case of the metrics coefficient of connectivity, density of plans, and decision density. However, in most cases, the aggregation of the metric has been computed as a weighted average. For example, in the case of the average connector degree metric, the aggregation has been calculated as a weighted average with respect to the number of connector nodes in each graph. Or, in the case of the depth metric, the weights are the number of nodes of every graph. However, the maximum connector degree has been obtained as the maximum over all maximum values.

The aggregation of the metrics in the partitionability category has been computed as a weighted average with respect to the number of nodes, except for the sequentiality metric where we have used as weight the sum of the number of nodes and the number of arcs. Finally, the aggregation of the metrics connector mismatch and control flow complexity in the connector interplay category and the metric in the concurrency category have been computed as weighted averages with respect to the number of split connector nodes in each graph.

Table 2 presents the values of the metrics for a PROforma CIG for the assessment and treatment of asthma. The second column displays the values for the top-level plan shown in Figure 1, i.e., single-graph metrics. The third column presents the values of the full-model metrics.

Table 2. Metric values for the Asthma CIG: the first column (*Graph*) shows the results for the top-level plan in Figure 1, while the second column (*Model*) shows the results for the full-model metrics. NB: BPM metrics reinterpreted for PROforma are marked with (*), the rest are PROforma-specific metrics.

Metric	**Graph**	**Model**
Size * S_N	7	46
Number of arcs * S_A	6	44
Diameter * $diam$	4	5.15
Number of plans S_P	3	10
Density * Δ	0.14	0.16
Coefficient of connectivity * CNC	0.86	0.51
Average connector degree * $\overline{d_C}$	3.00	2.79
Maximum connector degree * $\widehat{d(c)}$	3	4
Density of plans $\overline{p}$	0.43	0.22
Percentage of single-node plans $p1\%$	n/a	0.27
Average size of a plan $\bar{t}$	n/a	4.18
Percentage of plans of size above average $p50\%$	n/a	0.27
Decision density δ_D	0.29	0.17
Separability * Π	0.6	0.38
Sequentiality * Ξ	0.17	0.16
Structuredness * ϕ	1	0.73
Depth * Λ	2	1.43
Model depth Υ	n/a	5
Connector mismatch * MM	2	0.61
Control flow complexity * CFC	6	10
Number of decisions S_D	2	13
Number of preconditions $S_{Precond}$	4	19
Token split * TS	2	4.62

3. Experiments

3.1. Experimental Setting

As mentioned before, CIG models must be easy to maintain, to incorporate changes or to correct errors. Their design must facilitate their comprehension by modelers enabling them to easily find missing aspects or to include new findings. According to ISO 25010 quality model [19], modifiability and analyzability are two of the sub-characteristics of maintainability defined to assess the quality of systems and software products. Therefore, we have selected these characteristics to assess the quality of CIG models represented using the PROforma language. In this light, we have conducted an experiment to determine whether the proposed metrics are in correlation with the analyzability and modifiability of PROforma models and, consequently, whether these metrics can be used as quality indicators of PROforma models.

To design the experiment, we have considered different works focused on the assessment of metrics for modeling languages from other domains, such as Software Process Models [21], Entity–Relationship Diagrams [32], or Business Process Models [22]. Specifically, we have used the experimentation method proposed in Wohlin [33]. We have not considered works related to metrics for assessment of source code quality due to the evident differences between imperative programming and CIGs in PROforma. The next subsections describe the steps followed to prepare and carry out the experiment.

3.1.1. Scoping

Scoping includes the definition of the experiment and the identification of its goals. Regarding the experiment definition, we have defined a set of exercises of different types and varying complexity, to be solved by modelers, as done in similar works [22,32]. The exercises are related to PROforma CIGs based on guidelines from different medical specialties and developed by independent teams, to increase the diversity in the sample. The goal of the experiment has been defined similarly to the Goal/Question/Metric Method by

Basili [34]: *to assess PROforma metrics, with the purpose of validating if they report on quality indicators with respect to analyzability and modifiability, from the point of view of modelers, and in the context of modellers with different skill levels.*

3.1.2. Planning

Planning consists in identifying the subjects, preparing the material, specifying how the experiment is going to be performed and formulating the hypothesis. The subjects participating in the experiments were 13 people with different expertise profiles in the field of PROforma modeling, including members of our research group and graduate students, all of them with an IT background. The material used includes both the PROforma models and the exercises. Table 3 lists the PROforma models used. Except for the CHF and COPD ones, which were developed by our research group, the models come from the OpenClinical.net repository [30] and have been developed by independent modellers. Most of the models are based on guidelines from well-recognized international organizations (see *Source guideline* column in Table 3). Thus, our experiment included guidelines developed by a wide range of medical institutions: the British Thoracic Society (BTC), the Scottish Intercollegiate Guidelines Network (SIGN), the European Society of Cardiology (ESC), the Global Initiative for Chronic Obstructive Lung Disease (GOLD), the American College of Chest Physicians (ACCP), the UK National Institute for Health and Care Excellence (NICE), the New England Medical Center from the US (NEMC), and the Accident Compensation Corporation from New Zealand (ACC).

Table 3. PROforma models used in the experiment.

Model	Goal	Source Guideline	Size
Asthma	Assessment and treatment of asthma in adults and children	BTS/SIGN (UK)	46
CHF	Diagnosis and treatment of chronic heart failure	ESC (EU)	89
COPD	Diagnosis, management, and prevention of chronic obstructive pulmonary disease	GOLD (worldwide)	57
Cough	Diagnosis and treatment of chronic cough	ACCP (US)	28
CRcaTriage	Colorectal referral and diagnostic	NICE (UK)	7
Depression	Management of depression in primary care	NEMC (US)	18
Dyspepsia	Differential diagnosis of dyspepsia	N/A	4
HeadInjury	Work-up and management of acute head injury	NICE (UK)	34
IBME_TB	Screening for tuberculosis	unknown	14
Statins	Management of patients at elevated risk of coronary heart disease using statins	NICE (UK)	24
STIK	Assessment, investigation and management of soft-tissue injury of the knee	ACC (NZ)	26

The exercises consisted in questions or tasks, specifically defined for each PROforma model, to be solved by the subjects. The exercises were designed to reflect the analyzability and modifiability characteristics; therefore, they were classified into the *Analysis* and *Modifiability* categories (*Type*). Analysis exercises are those dealing with understanding the logical structure and/or the dynamic behavior of the model, and are usually formulated in terms of questions, e.g., about the results of model execution. Modifiability exercises are those where specific model changes are requested based upon a set of requirements. As mentioned before, it was intended that the exercises had different levels of complexity. Accordingly, we agreed that the exercises should fall in the categories *Trivial*, *Average*, and *Substantial* (*Difficulty*). Once the subjects have solved the exercises, the solutions must be rated. To this end, we initially planned to use the grades Correct (C), Partially Incorrect (PI), in case of minor flaws, or Incorrect (I).

As an illustration of the exercises designed, below are given an analyzability exercise and a modifiability one, both related to the top-level plan of Asthma CIG (see Figure 1). The two exercises were classified as trivial (T) in complexity:

Asthma CIG, worksheet #2, exercise #1, part (a)

> *In the Top-level Plan, would it be possible to omit the decision "Age Decision" and arrange the rest of the tasks so that the overall behavior of the plan remains the same?*

Asthma CIG, worksheet #2, exercise #1, part (b)

> *If so, modify the plan accordingly and make sure that the execution traces are compatible with the ones obtained before the changes.*

In the specification of how the experiment was going to be performed, we determined that, among the subjects, the researchers also had to participate preparing materials and exercises in the planning phase. Each participating researcher formulated different sets of exercises (or worksheets), most of them based on a specific PROforma model. The preparation of the worksheets included not only the wording of the exercises but also their classification in terms of type and complexity. These aspects were reviewed and validated by the most experienced modelers of the group. A total of 117 different exercises were formulated, grouped in 18 worksheets with 5–6 exercises each on average, with the exception of one worksheet which comprised 17 exercises.

Additionally, we determined that each subject had to participate in the operation phase solving the exercises proposed by the other participants and rating the solutions of the ones she/he had proposed. Needless to say, neither the participants answered the exercises they proposed, nor rated their own solutions to the exercises by others. More details of the operation step are included in the next subsection.

Finally, taking into account the goal of the experiment, the null hypothesis and its alternative were defined as follows:

- Null hypothesis H_0: there is no significant correlation between the metrics and the correct solutions in solving the exercises.
- Alternative hypothesis H_1: there is a significant correlation between the metrics and correct solutions in solving the exercises.

As with any experiment, there are some potential sources of bias that we have tried to minimize. Our results may be influenced by a selection bias due to the subjects and the PROforma models selected. The expertise in PROforma of the involved subjects may have an influence in the complexity of the exercises they prepare, including their classification and the grading of exercise solutions. To minimize this problem, subjects with different expertise levels were involved in these tasks. The fact that all the participants have a technical background might also be seen as a bias. However, we consider that this background should not be regarded as a bias as long as they have had the PROforma training required for the experiment. Finally, regarding PROforma models, their uniformity may also be a risk. To mitigate this bias, we have chosen PROforma models engineered by different people, and from different sources and medical specialities.

3.1.3. Operation

The operation phase of the experiment is composed of three main tasks: preparation, execution, and data validation. The preparation of the experiment includes all the actions required to have the material and participants ready. This comprised the instructions for the participants (including the wording of the exercises), the guideline models, the answering forms, and the tools (spreadsheets) for gathering and grading the answers provided.

Among the preparation tasks, we can consider the development of software tools to compute the values of the metrics for all the PROforma models used in the experiment. For this purpose, we have implemented a Java program that takes as input a PROforma file in XML format, transforms it into a series of graphs, and uses these graphs to compute the values of all the metrics proposed in this paper. Notice that, originally, the format of PROforma CIGs is plain text. To facilitate the processing of PROforma files, we have opted for building an appropriate metamodel using the Eclipse Modeling Framework (EMF) [35]. This solution has the advantage of providing code generation facilities, such as editing tools adapted to the metamodel. We have used such an editor to produce the PROforma XML files required to compute the metric values.

Regarding the execution of the experiment, as mentioned before, all participants played the subject role, solving the exercises proposed by the rest of participants. The models were displayed and handled using the PROforma graphical editor Tallis Composer [36]. The solutions to the exercises were recorded in the provided answering forms and, if required, in modified PROforma models. Moreover, the subjects screen-recorded themselves while they were solving the exercises, describing the steps followed talking aloud. After reviewing the answers recorded in the forms, the modified models, and the screen-recorded videos, each participating researcher graded the exercises she/he had proposed as Correct (C) or Incorrect (I) using as tool the spreadsheets provided. Note that due to the inherent difficulty of establishing objective criteria to grade an exercise as partially incorrect, we finally opted for a binary rating of the exercises as Correct (C) or Incorrect (I).

The final task consists in checking that the data collected in the above spreadsheets is valid. First, we have removed invalid and void answers, e.g., answers stating that the exercise had no solution or that the wording made no sense. Second, and related to the latter, we have removed those exercises with severe ambiguities in their wording. To do this, we have considered a wording as ambiguous when three or more subjects had concerns about it. After this validation, we obtained a total of 368 observations, each one corresponding to an exercise solution by one subject, and including the rating of the solution, as well as the exercise and subject details.

3.2. Results

In this section, we describe the last step of the experiment. We explain the statistical analysis performed to validate the hypothesis and we discuss the results obtained. As described in Section 2.2, we have two sets of metrics regarding their scope: single-graph metrics and full-model metrics. Therefore, we have performed two distinct analyses. The variables included in each statistical analysis are the calculated values for all the metrics in the graphs or model involved in the exercise, an anonymous identifier of the subject who solved the exercise, and the level of difficulty of the exercise. Additionally, the type of the exercise was taken into account by considering in the statistical test three different poolings of exercises: (1) all the exercises, (2) only the analysis exercises, and (3) only the modifiability exercises. The observations correspond to the correctness, i.e., the correct or incorrect grading of the exercise solution. We hypothesized that the correlation between the metric and the correctness of the exercise solution will depend on the type of the exercise.

For the statistical analyses, we have computed the correlation between two variables: (1) the interaction between the metrics with the difficulty of the exercise ($Metric \times Difficulty$) and (2) the correctness of the exercise. For that purpose, we have selected the generalized linear mixed model with the binary logistic regression (GLMM-BLR) [37] as target distribution. On the one hand, the binary logistic regression has been selected since the output, the exercise grade or correctness, falls into two non-ordinal categories: correct and incorrect. On the other hand, the mixed models are suitable when data include correlated or non-independent observations. In this case, we have several observations from the same subject. Thus, we have used the previously mentioned interaction as the fixed effect in the mixed model and the variable *person*—the subject who answered the exercise—as the random effect in the mixed model. As a result, we have reformulated the hypothesis definition as follows:

- Null hypothesis H_0: there is no significant correlation between the interaction of metrics and Difficulty, and correct solutions in solving the exercises.
- Alternative hypothesis H_1: there is a significant correlation between the interaction of metrics and Difficulty, and correct solutions in solving the exercises.

Section 3.2.1 is devoted to the results of the analysis using the metrics for single graphs, whereas Section 3.2.2 focuses on the metrics for the full model.

3.2.1. Analysis of Single-Graph Metrics

In this first scenario, the analysis has been performed using the metrics calculated for the graph involved in each exercise and for the three above mentioned poolings of exercises. The fixed effect in the GLMM-BLR model is the interaction between the single-graph metrics and the difficulty of the exercise ($Metric \times Difficulty$). Table 4 summarizes the results of the metrics whose significance level was below the threshold value ($\rho < 0.05$) in the statistical test and, thus, showing a statistically significant correlation in, at least, one of the three above mentioned poolings. Henceforth, when we use the term "statistically significant correlation", we mean that the metric is relevant and its values can be regarded as an indicator of the difficulty on solving exercises. The observed outcomes are summarized below.

Table 4. Single-graph metrics showing a significance level for the fixed effect when considering: (1) all exercises (All), (2) analysis exercises only (Ana.), and (3) modifiability exercises only (Mod.). The significant values are highlighted in bold text.

Category	Metric & Fixed Effect	All	Ana.	Mod.
Size	Size: $S_N \times Difficulty$	**0.013**	**0.011**	0.195
	Number of arcs: $\|A\| \times Difficulty$	**0.031**	**0.001**	0.205
	Diameter: $diam \times Difficulty$	**0.033**	**0.014**	0.133
Density	Density: $\Delta \times Difficulty$	**0.041**	0.054	**0.027**
	Coefficient of connectivity: $CNC \times Difficulty$	**0.020**	**0.018**	**0.041**
	Average connector degree $\overline{d_C} \times Difficulty$	**0.014**	**0.011**	0.082
	Maximum connector degree $\widehat{d_C} \times Difficulty$	**0.021**	**0.007**	0.138
Partitionability	Separability: $\Pi \times Difficulty$	0.055	0.238	**0.015**
	Sequentiality: $\Xi \times Difficulty$	**0.044**	0.204	0.148
	Depth: $\Lambda \times Difficulty$	0.090	**0.043**	0.111
Conn. interplay	Control flow complexity: $CFC \times Difficulty$	0.08	**0.041**	0.570
	Number of decisions: $S_D \times Difficulty$	0.059	**0.033**	0.398
Concurrency	Token split: $TS \times Difficulty$	0.086	**0.022**	0.513

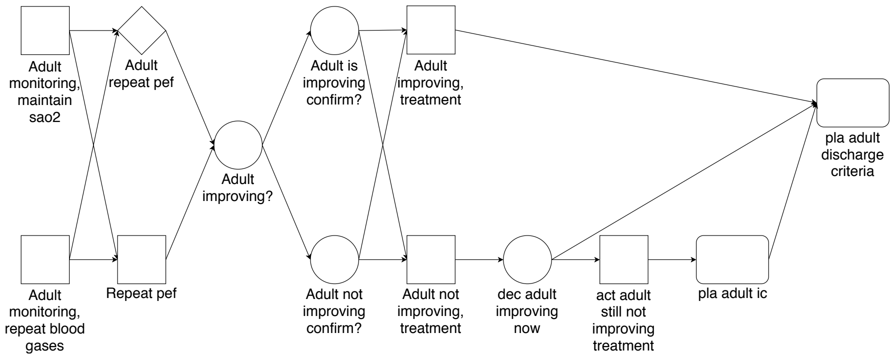
First, the subset of metrics under the categories size and density have a significant correlation for the first pooling of exercises (all exercises). They also have a significant correlation for the second pooling (analysis exercises), except for density (Δ) metric, whose significance value is close but slightly above the threshold of 0.05. However, they do not usually show a significant correlation for the third pooling (modifiability exercises). In fact, the metrics that indicate a ratio between the number of arcs and the number of nodes, namely, density (Δ) and coefficient of connectivity (CNC), are the only ones showing a significant correlation for that case.

Second, the subset of metrics under the remaining categories (partitionability, connector interplay, and concurrency) show a less clear pattern in the three poolings of exercises. The only metric showing a significant correlation for all exercises is sequentiality (Ξ), which somehow equates the simplicity of the graph to the presence of sequences of nodes. The metrics showing a significant correlation for the analysis exercises are depth (Λ), control flow complexity (CFC), number of decisions (S_D), and token split (TS), which are all related to connectors and possible execution paths. The only metric showing a significant correlation for the modifiability exercises is separability (Π), which might indicate the presence of non-trivial structures in the graph.

In general, for the analysis exercises, it makes sense that errors occur in those graphs with a large number of elements, specially those including many multiple-path decisions.

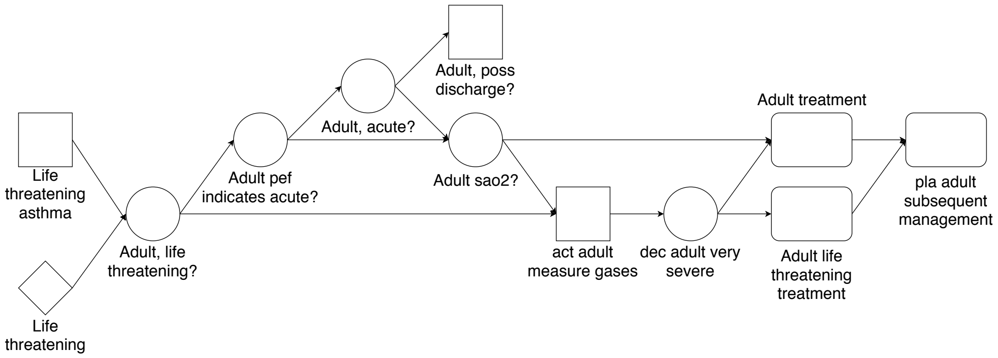
In such cases, all the tasks and scheduling constraints have to be analyzed in order to understand the structure of the model or its behavior.

In contrast to analysis exercises, the modifiability ones request specific changes which in some cases do not require the understanding of the complete graph, e.g., because the focus is in a small part of it. Therefore, those metrics indicating the global size of the graph might not show a significant correlation for the modifiability exercises. The understandability for this kind of exercises does not depend on the number of elements in the graph, but on the complexity of the nodes involved in the modification. However, a simple modification in a dense model, such as the two examples shown in Figures 3 and 4, might require a comprehensive understanding of all the tasks in the model because complex elements (such as preconditions and triggers) might be used for synchronization purposes. The understandability issues raised in these two examples are not derived from their size, but because of the complex structures and relationship between tasks. Moreover, the complexity of the structures in these examples is somehow captured by the metrics showing the ratio between the scheduling constraints and the number of nodes—density (Δ) and coefficient of connectivity (CNC)—and the metrics analyzing the structure of the graph—separability (Π) and connector mismatch (MM). Note that the coefficient of connectivity (CNC) reaches values above 1 in poorly structured models and/or in those with some complex interactions among tasks, e.g., nodes with two or more in-coming arcs (see Figures 3 and 4). Thus, the more complex the graph involved in the exercise is, the more error-prone it is on modification tasks.


Adult monitoring, maintain sao2
Adult repeat pef
Adult is improving confirm?
Adult improving, treatment
pla adult discharge criteria
Adult improving?
Adult monitoring, repeat blood gases
Repeat pef
Adult not improving confirm?
Adult not improving, treatment
dec adult improving now
act adult still not improving treatment
pla adult ic

S_N	13	Π	0.09
$\lvert A \rvert$	18	Ξ	0.11
$diam$	8	Λ	1
Δ	0.12	MM	0
CNC	1.38	CFC	19
$\overline{d_C}$	3	S_D	4
$\widehat{d_C}$	4	TS	6

Figure 3. Graph "*adult subsequent management*" of Asthma CIG. The 66.67% of analysis exercises and the 66.67% of modifiability exercises were correctly solved.



S_N	12	Π	0.2
$\lvert A \rvert$	15	Ξ	0
$diam$	8	Λ	2
Δ	0.11	MM	1
CNC	1.25	CFC	17
$\overline{d_C}$	3.125	S_D	5
$\widehat{d_C}$	4	TS	5

Figure 4. Graph "*adult assessment*" of Asthma CIG. The 73.33% of analysis exercises and 66.67% of modifiability exercises were correctly solved.

3.2.2. Analysis of Full-Model Metrics

We have performed a second analysis using the full-model metrics calculated for the model involved in each exercise and, as in the previous scenario, for the three poolings of exercises. Except for the metrics, the statistical analysis considers the same variables: fixed effect and random effect. Table 5 summarizes the results of the metrics whose significance level was below the threshold value ($\rho < 0.05$) in the statistical test and, thus, showing a statistically significant correlation in, at least, one of the three above mentioned cases. The observed outcomes are summarized below.

In this case, most of the metrics (14 out of 16) fall under the categories density and partitionability. For the metrics under the size category, diameter ($Diam$) is the only one showing a significant correlation for the pooling with all exercises. Note that the metrics under this category usually compute the aggregation as the sum of the metric over all the graphs in the model, except diameter ($Diam$) which applies a weighted average. For the metrics falling under the connector interplay category, only the number of preconditions ($S_{Precond}$) metric is showing a statistical correlation and just for the analysis exercises. None of the full-model metrics under the concurrency category show a significant correlation.

All the metrics within the density category are included in the table, which may demonstrate that densities are of special relevance. This is even more apparent in the modifiability exercises, where seven out of the nine metrics within the density category show a significant correlation in one or more of the three poolings of exercises. It is of special relevance that the three full-model metrics we specifically defined for PROforma models—percentage of single node plans ($p1\%$), average size of a plan ($\bar{t}$), and percentage of plans whose size is over the average size ($p50\%$)—are significant or show a significance value slightly higher than the threshold, except for the percentage of single node plans ($p1\%$) in modifiability exercises.

Almost all the metrics within the partitionability category indicate a significant correlation for the pooling including all the exercises. The metrics depth (Λ) and model depth (Υ) also show a significant correlation for the pooling including the analysis exercises, whereas the separability (Π), sequentiality (Ξ), and structuredness (ϕ) also show a significant correlation for the last pooling of exercises, the one with the modifiability exercises. Only the metric structuredness (ϕ) shows a significant correlation in the three poolings of exercises. This suggests that the metrics capturing the complexity on the structure of the full model, such as separability (Π), sequentiality (Ξ), and structuredness (ϕ), may somehow indicate how error-prone the model is in the modifiability exercises, whereas the ones indicating the depth are of special relevance for the analysis exercises.

Table 5. Full-model metrics showing a significance level for the fixed effect when considering: (1) all exercises (All), (2) analysis exercises only (Ana.), and (3) modifiability exercises only (Mod.). The significant values are highlighted in bold text.

Cat.	**Metric & Fixed Effect**	**Alls**	**Ana.**	**Mod.**
Size	Diameter: $Diam_{aggr} \times Difficulty$	**0.014**	0.050	0.055
Density	Density: $\Delta_{aggr} \times Difficulty$	0.296	0.212	**0.032**
	Coef. of connectivity: $CNC_{aggr} \times Difficulty$	**0.008**	0.114	**0.025**
	Avg. connector degree: $\overline{d_{C_{aggr}}} \times Difficulty$	**0.012**	**0.032**	**0.030**
	Max. connector degree: $\widehat{d_{C_{aggr}}} \times Difficulty$	0.217	0.093	**0.027**
	Density of plans: $\overline{p} \times Difficulty$	0.068	**0.021**	0.095
	Perc. single node plans: $p1\% \times Difficulty$	**0.031**	**0.039**	0.374
	Avg. size plan: $\overline{t} \times Difficulty$	**0.007**	**0.026**	**0.043**
	Perc. plans over average: $p50\% \times Difficulty$	**0.002**	0.061	**0.038**
	Decision density: $\delta_D \times Difficulty$	**0.005**	**0.005**	**0.044**
Partitionability	Separability: $\Pi_{aggr} \times Difficulty$	**0.016**	0.087	**0.030**
	Sequentiality: $\Xi_{aggr} \times Difficulty$	**0.017**	0.090	**0.048**
	Structuredness: $\phi_{aggr} \times Difficulty$	**0.037**	**0.031**	**0.043**
	Depth: $\Lambda_{aggr} \times Difficulty$	**0.026**	**0.032**	0.051
	Model depth: $\Upsilon \times Difficulty$	0.119	**0.047**	0.151
Conn. Interplay	Number of preconditions: $S_{Predcond} \times Difficulty$	0.304	**0.042**	0.204

4. Discussion

The previous results show that 13 single-graph and 16 full-model metrics are statistically significant in that they report on the complexity of the PROforma models. It is interesting to remark that four out of the 16 full-model metrics deemed significant, namely, percentage of single node plans ($p1\%$), average size plan ($\overline{t}$), percentage of plans over average ($p50\%$), and structuredness (ϕ_{aggr}), can only be calculated over the full model, i.e., the corresponding single-graph metric does not exist. Then, the global number of metrics deemed significant is similar when the analyses are limited to those metrics that have a definition for the single graph and the full model.

Among the relevant metrics, the ones showing a significance level below 0.05, only eight of them are statistically significant in both analyses, namely, density (Δ), coefficient of connectivity (CNC), average connector degree ($\overline{d_C}$), maximum connector degree ($\widehat{d_C}$), separability (Π), sequentiality (Ξ), and depth (Λ). In other words, these eight metrics are statistically significant when they are calculated over the single graph but also when they are calculated over the full model with the aggregation formula. In most of cases (12 out of the 16 cases (eight metrics × two analyses)), the metrics show a significant correlation for the pooling including all exercises. However, the behavior is different for the other two poolings of exercises.

For the pooling including only the analysis exercises, the statistical results suggest that the relevant metrics fit better when they are calculated over single graphs, as five out of eight single-graph metrics are statistically significant. In contrast, only two out of eight full-model metrics are statistically significant. For the pooling including only the modifiability exercises, the statistical results suggest the opposite, i.e., the relevant metrics fit better when they are calculated over the full model. For this kind of exercises, three out of eight single-graph metrics are statistically significant, whereas seven out of eight full-model metrics have a significance value below the threshold of 0.05. This might indicate that the results on the analysis exercises are more sensitive to the complexity of the single graph, whereas the results on the modifiability exercises have higher dependence on the complexity of the full model.

When the analysis only includes modifiability exercises, the statistical results do not show a clear correlation between the single-graph metrics and the exercise grades. There are only three out of 13 cases (less than 23%) where the single-graph metrics are statistically significant. In contrast, full-model metrics might be useful for that particular type of exercises. In 10 out of 16 cases (62.5%), the metrics are statistically significant and the statistical significance is sightly above the threshold in two cases (12.5%). According to these 10 metrics, the more complex the model is, the more error-prone it is on modification tasks.

Finally, the global number of metrics deemed as statistically significant is 21, with five of them being relevant for single graphs, eight of them relevant for full models, and eight of them relevant for both. For the single graph analysis, the metrics deemed as significant are roughly equally distributed over the five categories. In contrast, the analysis for the full-model metrics shows that the vast majority of the metrics deemed as significant (14 out of 16) are within just two categories, namely, density and partitionability. This finding was expected and the results have empirically indicated that modifying models with complex structures may be more prone to error.

5. Conclusions

In this research work, we aimed at providing a comprehensive set of metrics that allow to measure model quality aspects of PROforma CIG models. Although there are many proposals for quality assurance in the field of software and business process models, to the best of our knowledge there are no studies dealing with the quality of CIG models in terms of understandability and modifiability. First, we have proposed metrics inspired in BPM ones to include common process modeling characteristics. Second, we have considered specific features of PROforma to define new metrics that capture distinctive aspects of this CIG language. These include metrics that take into account the modularity and hierarchical decomposition aspects of PROforma models. Finally, we have carried out an empirical validation of these metrics as quality indicators. To this purpose, we have conducted an experiment and carried out a statistical analysis of its results that are presented as a part of this research work.

We can conclude that the metrics proposed in this paper can be used as indicators of the understandability, and thereby maintainability, of PROforma CIGs whenever the difficulty of the task is considered. We have observed that when single graphs are taken into consideration, practically all statistically significant metrics belong to the ones inspired in the BPM metrics. However, the newly defined metrics for PROforma prevail as statistically significant when the whole model is considered, which means that they better characterize its complexity. Another observation related to single-graph metrics is that the statistically significant metrics when analysis tasks are considered are disjoint from the significant metrics in the case of modifiability tasks, in most of the cases. Concretely, the single-graph metrics under the categories size, connector interplay, and concurrency appear to be good indicators when solving analysis exercises, whereas the single-graph metrics in the density category seem to be significant in solving modifiability exercises. In the case of the full-model metrics, and for modifiability tasks, the statistically significant metrics are distributed among density and partitionability categories. Still, many of the full-model metrics that we have proposed show correlation for some of the poolings.

In light of the above, we consider that the proposed metrics can be of great value, as they capture the PROforma-specific features in addition to those inspired by the general-purpose process model metrics in the literature. We believe that the metrics we have defined are generic enough, and thus we foresee that they could also be a valuable contribution to CIG languages other than PROforma. Notably, a number of metrics (including full-model ones) could be easily adapted and applied to CIG languages following the TNM approach, as PROforma does. As mentioned earlier, this approach is shared by many CIG languages. Furthermore, decision-related metrics could also be generalized, as PROforma's decision model has been adopted by several CIG languages [14].

One limitation of our work lies in the experimental setting, concretely in that the experiment has not particularly been large-scale considering the number of subjects and guideline models. Despite this, we consider that the results we have obtained constitute a valuable contribution from which further research can be undertaken. As future work, a more comprehensive experiment, including more subjects as well as more guidelines and exercises, could be devised to increase the reliability and stability of our results. Note that we do not consider that the lack of participation of clinician subjects in our study should be regarded as a limitation. In our view, we would have obtained a similar outcome if clinicians with the minimum PROforma training required had participated in the study. On the other hand, some additional PROforma elements that can affect the model understandability are not represented in the graph structure currently used as a basis for the analysis performed. Another line of future work will be focused on describing and implementing new metrics that consider those PROforma features, as they could bring new objective insights about the actual complexity of the model independently from its layout.

Author Contributions: Conceptualization, M.M. and B.M.-S.; methodology, M.M., B.M.-S., and C.C.S.; software, B.M.-S. and J.T.-S.; validation, J.T.-S., B.M.-S. and C.C.S.; formal analysis, J.T.-S., B.M.-S., and C.C.S.; writing—original draft preparation, J.T.-S., B.M.-S., C.C.S., and M.M.; writing—review and editing, J.T.-S., B.M.-S., C.C.S., and M.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been funded by the Spanish Ministry of Economy and Competitiveness through the grant TIN2014-53749-C2-1-R.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Institute of Medicine. *Clinical Practice Guidelines We Can Trust*; The National Academies Press: Washington, DC, USA, 2011.
2. Peleg, M. Computer-interpretable clinical guidelines: A methodological review. *J. Biomed. Inform.* **2013**, *46*, 744–763. [CrossRef] [PubMed]
3. Sonnenberg, F.A.; Hagerty, C.G. Computer-interpretable clinical practice guidelines. Where are we and where are we going? *Yearb. Med. Inform.* **2006**, 15, 145–158.
4. Patel, V.L.; Arocha, J.F.; Diermeier, M.; Greenes, R.A.; Shortliffe, E.H. Methods of Cognitive Analysis to Support the Design and Evaluation of Biomedical Systems: The Case of Clinical Practice Guidelines. *J. Biomed. Inform.* **2001**, *34*, 52–66. [CrossRef] [PubMed]
5. Mulyar, N.; van der Aalst, W.M.P.; Peleg, M. A Pattern-based Analysis of Clinical Computer-interpretable Guideline Modeling Languages. *J. Am. Med. Inform. Assoc.* **2007**, *14*, 781–787. [CrossRef] [PubMed]
6. Grando, M.A.; Glasspool, D.; Fox, J. A formal approach to the analysis of clinical computer-interpretable guideline modeling languages. *Artif. Intell. Med.* **2012**, *54*, 1–13. [CrossRef] [PubMed]
7. Kaiser, K.; Marcos, M. Leveraging workflow control patterns in the domain of clinical practice guidelines. *BMC Med. Inform. Decis.* **2016**, *16*, 1–23. [CrossRef] [PubMed]
8. Mendling, J.; Reijers, H.; van der Aalst, W. Seven process modeling guidelines (7PMG). *Inf. Softw. Technol.* **2010**, *52*, 127–136. [CrossRef]
9. Gruhn, V.; Laue, R. Approaches for Business Process Model Complexity Metrics. In *Technologies for Business Information Systems*; Abramowicz, W., Mayr, H.C., Eds.; Springer: Dordrecht, The Netherlands, 2007; pp. 13–24.
10. Marcos, M.; Torres-Sospedra, J.; Martínez-Salvador, B. Assessment of Clinical Guideline Models Based on Metrics for Business Process Models. In *Knowledge Representation for Health Care*; Springer: Cham, Switzerland, 2014; Volume 8903, pp. 111–120.
11. Mendling, J. Metrics for Business Process Models. In *Metrics for Process Models: Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 103–133.
12. Fox, J.; Johns, N.; Rahmanzadeh, A. Disseminating medical knowledge: The PROforma approach. *Artif. Intell. Med.* **1998**, *14*, 157–181. [CrossRef]

13. Greenes, R.A.; Bates, D.W.; Kawamoto, K.; Middleton, B.; Osheroff, J.; Shahar, Y. Clinical decision support models and frameworks: Seeking to address research issues underlying implementation successes and failures. *J. Biomed. Inform.* **2018**, *78*, 134–143. [CrossRef] [PubMed]
14. Peleg, M.; González-Ferrer, A. Chapter 16—Guidelines and Workflow Models. In *Clinical Decision Support. The Road to Broad Adoption*, 2nd ed.; Greenes, R.A., Ed.; Academic Press: Oxford, UK, 2014; pp. 435–464.
15. The AGREE Collaboration. Development and validation of an international appraisal instrument for assessing the quality of clinical practice guidelines: the AGREE project. *Qual. Saf. Health Care* **2003**, *12*, 18–23. [CrossRef] [PubMed]
16. Guyatt, G.H.; Oxman, A.D.; Vist, G.E.; Kunz, R.; Falck-Ytter, Y.; Alonso-Coello, P.; Schünemann, H.J. GRADE: An emerging consensus on rating quality of evidence and strength of recommendations. *BMJ* **2008**, *336*, 924–926. [CrossRef] [PubMed]
17. Rushby, J. *Quality Measures and Assurance for AI Software*; Technical Report 4187; SRI International, Computer Science Laboratory: Menlo Park, CA, USA, 1988.
18. Miguel, J.P.; Mauricio, D.; Rodríguez, G. A Review of Software Quality Models for the Evaluation of Software Products. *Int. J. Softw. Eng. Appl.* **2014**, *5*, 31–54. [CrossRef]
19. ISO. *ISO/IEC 25010:2011 Systems and Software Engineering—Systems and Software Quality Requirements and Evaluation (SQuaRE)—System and Software Quality Models*; Technical Report; ISO International Organization for Standardization: Geneva, Switzerland, 2011.
20. Maxim, B.R.; Kessentini, M. Chapter 2—An introduction to modern software quality assurance. In *Software Quality Assurance in Large Scale and Complex Software-Intensive Systems*; Morgan Kaufmann: Burlington, MA, USA, 2016; pp. 19–46.
21. Canfora, G.; García, F.; Piattini, M.; Ruiz, F.; Visaggio, C.A. A family of experiments to validate metrics for software process models. *J. Syst. Softw.* **2005**, *77*, 113–129. [CrossRef]
22. Sánchez-González, L.; García, F.; Mendling, J.; Ruiz, F. Quality Assessment of Business Process Models Based on Thresholds. In *On the Move to Meaningful Internet Systems: OTM 2010—Part I*; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6426, pp. 78–95.
23. Hasić, F.; Vanthienen, J. Complexity metrics for DMN decision models. *Comput. Stand. Interfaces* **2019**, *65*, 15–37. [CrossRef]
24. Moody, D.L. Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. *Data Knowl. Eng.* **2005**, *55*, 243–276. [CrossRef]
25. ten Teije, A.; Marcos, M.; Balser, M.; van Croonenborg, J.; Duelli, C.; van Harmelen, F.; Lucas, P.; Miksch, S.; Reif, W.; Rosenbrand, K.; et al. Improving medical protocols by formal methods. *Artif. Intell. Med.* **2006**, *36*, 193–209. [CrossRef] [PubMed]
26. Hommersom, A.; Groot, P.; Lucas, P.; Marcos, M.; Martínez-Salvador, B. A Constraint-Based Approach to Medical Guidelines and Protocols. In *Computer-Based Medical Guidelines and Protocols: A Primer and Current Trends*; Studies in Health Technology and Informatics; Springer: Berlin/Heidelberg, Germany, 2008; Volume 139, pp. 213–222.
27. Sutton, D.R.; Fox, J. The Syntax and Semantics of the PROforma Guideline Modeling Language. *J. Am. Med. Inform. Assoc.* **2003**, *10*, 433–443. [CrossRef] [PubMed]
28. Object Management Group (OMG). Case Management Model and Notation. 2016. Available online: https://www.omg.org/spec/CMMN (accessed on 29 April 2020).
29. Object Management Group (OMG). Decision Model and Notation Version 1.2. 2019. Available online: https://www.omg.org/spec/DMN/1.2 (accessed on 29 April 2020).
30. OpenClinical CIC. OpenClinical.net. 2016. Available online: https://www.openclinical.net/ (accessed on 29 April 2020).
31. Reijers, H.A.; Mendling, J. Modularity in Process Models: Review and Effects. In Proceedings of the 6th International Conference on Business Process Management (BPM 2008), Milan, Italy, 2–4 September 2008; pp. 20–35.
32. Genero, M.; Poels, G.; Piattini, M. Defining and validating metrics for assessing the understandability of entity–relationship diagrams. *Data Knowl. Eng.* **2008**, *64*, 534–557. [CrossRef]
33. Wohlin, C.; Runeson, P.; Höst, M.; Ohlsson, M.C.; Regnell, B.; Wesslén, A. *Experimentation in Software Engineering: An Introduction*; International Series in Software Engineering; Springer: Berlin/Heidelberg, Germany, 2000.
34. Basili, V.R. *Software Modeling and Measurement: The Goal/Question/Metric Paradigm*; Technical Report, Techreport UMIACS TR-92-96; University of Maryland: College Park, MD, USA, 1992.
35. Steinbrg, D.; Budinsky, F.; Patenostro, M.; Merks, E. *EMF: Eclipse Modeling Framework*; Addison Wesley: Boston, MA, USA, 2008.
36. COSSAC IRC in Cognitive Science & Systems Engineering. Tallis. 2007. Available online: http://openclinical.org/tallis.html (accessed on 14 May 2019).
37. Agresti, A. *Categorical Data Analysis*, 3rd ed.; Wiley: Hoboken, NJ, USA, 2012.