Journal of Computer Languages 84 (2025) 101328

Contents lists available at ScienceDirect

Journal of Computer Languages

journal homepage: www.elsevier.com/locate/cola

Towards democratisation of veterinary clinical protocols: Transferring their development from technical-coding experts to veterinary professionals for the case of Chronic Kidney Disease for Cats (CKD4Cats Domain-Specific Language)

Sofia Meacham [a,*], Hessa Alfraihi [b]

[a] Computing & Informatics Department, Bournemouth University, Fern Barrow, Poole, BH12 5BB, United Kingdom
[b] Department Of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia

ARTICLE INFO

Dataset link: https://github.com/stsasakou/CKDprotocol4cats/tree/main

Keywords:
Domain-Specific Languages
Clinical protocols
Veterinary Informatics
Chronic Kidney Disease for cats guidelines
Democratisation of technology

ABSTRACT

This paper presents CKD4Cats, a domain-specific language (DSL) for computerised Chronic Kidney Disease (CKD) clinical protocols in cats - a very common disease in veterinary practice. Building on DSLs used in human health, CKD4Cats addresses veterinary-specific needs while addressing their shortcomings. Developed with JetBrains' Meta-Programming System (MPS) and veterinary input, the DSL ensures ease of use and adoption. It employs advanced evaluation methods, creating a projectional editor that streamlines protocol creation, displays relevant options, and guarantees "correct-by-construction" clinical protocols. This innovative approach democratises software development, making advanced tools accessible to non-technical users and significantly improving veterinary practice management.

1. Introduction

Chronic Kidney Disease (CKD) in cats represents a significant and prevalent condition within veterinary medicine, presenting a continual challenge for both new graduates and seasoned veterinary practitioners. As the prevalence of CKD increases notably in ageing feline populations, with rates escalating to as high as 30%–40% in cats over 10 years old [1,2], the need for accurate diagnosis, effective monitoring, and appropriate treatment becomes increasingly critical. The ongoing research in this field highlights not only the complexity of managing CKD but also the necessity for innovative tools and approaches to enhance clinical practice. A veterinary information system that aids the diagnosis and management of CKD for cats in a quick, reliable, and scientifically correct way, is a helpful tool that could support the monitoring of cats with CKD and improve their well-being.

On the other hand, Clinical Protocols, also known as Clinical Practice Guidelines (CPGs), have been integral to medical practices since 1990. According to [3], they are "systematically developed statements to assist practitioner and patient decisions about appropriate care for specific clinical circumstances". While their implementation varies, with most practices using them as advisory documents, CPGs can be referenced in court as expert testimony in cases of alleged malpractice [4]. In technology, clinical protocols are commonly used within Computerised Decision Support Systems (CDSS), serving as the rules that ensure "correctness" by complying with official guidelines. Research in human medicine has extensively examined the use of clinical protocols in CDSS [5–7], demonstrating that software modelling of these protocols can enhance clinical decision-making for both experienced doctors and general practitioners [8].

Given the complexities of managing CKD in cats, integrating computerised clinical protocols into veterinary information systems could similarly ensure adherence to best practices. If properly developed and validated, these protocols can help standardise care, improve diagnostic accuracy, and ultimately enhance the quality of treatment provided to feline patients.

The development and adoption of computerised clinical protocols, particularly through domain-specific languages (DSLs), has been extensively explored in human medicine. These protocols, supported by DSLs, offer a structured approach to clinical decision-making, enhancing both the accuracy and consistency of care [9,10]. However, veterinary medicine has yet to fully embrace similar advancements, particularly in the creation and utilisation of DSLs for clinical protocols. The potential for such technologies in veterinary medicine, especially

* Corresponding author.
E-mail addresses: smeacham@bournemouth.ac.uk (S. Meacham), haalfraihi@pnu.edu.sa (H. Alfraihi).

https://doi.org/10.1016/j.cola.2025.101328
Received 17 November 2024; Received in revised form 17 March 2025; Accepted 2 May 2025
Available online 18 May 2025

for managing conditions like CKD in cats, represents a promising area of exploration that could significantly enhance clinical outcomes and standardise care practices across the field.

This study aims to bridge the gap between human and veterinary medicine by developing a projectional DSL specifically tailored to veterinary clinical protocols, with a focus on CKD in cats. By leveraging the robust features of MPS JetBrains [11], this DSL is designed to address the unique challenges of veterinary practice, ensuring ease of use, formal correctness, and enhanced adoption among veterinary professionals. Through this approach, we aim to advance the shortcomings of DSLs for human clinical protocols and achieve to democratise advanced software development (projectional DSL) to non-technical domain users-veterinarians.

A background study on CKD for cats, computerised clinical protocols for human and veterinary medicine and MPS JetBrains DSL as a solution is presented in Section 2. In Section 3, our proposed solution through the development of a DSL will be given in detail. A case study was used to demonstrate the proposed language design and development process: the chronic kidney disease for cats' clinical protocol. The DSL developed will be described with two main subsections: language structure and language usage. In Section 4, language engineers' and veterinary professionals' evaluation will be presented. Finally, our conclusions and plans will be detailed in Section 5.

2. Background study

2.1. Chronic kidney disease for cats

Chronic kidney disease (CKD) in cats represents a very common but still challenging pathologic condition to diagnose and monitor, equally for new graduates and experienced veterinary practitioners. The overall prevalence of CKD in cats is 2% to 4%, and increases to 30% to 40% in cats over 10 years of age [1,2,12], so it is a condition frequently occurring in practice and concerns many cat owners. Therefore, the research attracts clinical and could attract public or even charity interest.

CKD is a chronic disease and cats tend to have a relatively good quality of life and for a considerable amount of time after diagnosis. Therefore, it requires follow-up and monitoring closely [2,13,14]. The owners visit veterinarians often to regularly assess the stage of the disease and adjust medications and treatment. As a result, a veterinary information system that aids the diagnosis in a quick, reliable, and scientifically correct way, is a helpful tool that could support the monitoring of cats with CKD and improve their well-being.

The corresponding clinical protocol guidelines are issued by the International Renal Interest Society (IRIS) [15] and provide comprehensive guidance to veterinary professionals, so they can monitor cats and tailor their treatment according to laboratory and clinical findings. It is interesting to notice that the research in this field never stopped growing besides the fact that CKD is an "old" pathological condition. Innovations in feline clinical diets and diagnostic tools such as the Symmetric dimethylarginine (SDMA) parameter [16] or RenalTech® [17], are creating a necessity to constantly update and review treatments and monitoring tools for cats with CKD. As such, IRIS updated its guidance again in 2023, although the previous guidance was published in 2019 (IRIS 2023a, IRIS 2023b).

CVS Group plc [18], a leading integrated veterinary services provider in the UK, established in 1999, acknowledges the importance and prevalence of the disease, so includes detailed resources and flowcharts on how to approach the condition on its Knowledge Hub network [19].

To elevate these approaches beyond merely advisory documents, greater rigour is needed. Developing computerised clinical protocols would bring the guidelines to the next level, ensuring more consistent and reliable implementation.

2.2. Computerised clinical protocols and DSLs: human and veterinary medicine

In human medicine, there is extensive research on computerised clinical protocols and DSLs. The research works for computerised clinical protocols in human health and DSLs has used a range of DSLs that were developed for this purpose. The DSLs lay in two groups: those with formal underpinnings and those without.

The most widely used domain-specific languages (DSLs) for clinical protocols with formal underpinnings include PROforma, Guideline Definition Language (GDL), MediK, event-B, Arden Syntax, Asbru, and GLIF. As part of our MDEnet-funded project, an EPSRC-supported network for Model-Driven Engineering, we evaluated PROforma and GDL, with findings presented in [20]. PROforma, developed in the 1990s by John Fox [21], has been applied in various medical contexts in the UK, such as the CAPSULE system for assisting General Practitioners. While its visual design aids usability, its reliance on formal methods and programming expertise can be a barrier [10,21]. PROformajs [22], a JavaScript-based engine, was required to simplify integration, but the graphical interface ultimately proved intuitive for medical professionals. Similarly, DSLs like Arden [23], Asbru [24], and GLIF [25] share PROforma's formal foundation but face similar challenges in usability for non-technical users. The Guideline Definition Language (GDL) leverages UML metamodeling for rule-based modelling and integrates with model-driven engineering solutions. Unlike other DSLs, GDL is aligned with the openEHR standard [26], promoting standardised health information. However, its rigidity and complexity pose challenges for adaptability in dynamic clinical environments [27,28]. Our MDEnet project revealed that archetype instantiation in GDL introduced significant complications. MediK, with its model checker, symbolic execution engine, and deductive verifier, ensures precise, executable guidelines but is resource-intensive [29]. Event-B approaches, integrating domain knowledge with formal proofs, have been utilised to improve guideline quality at a system level [30]. While formal methods offer precision and verifiability, they are resource-heavy and have a steep learning curve, making them less practical for rapidly changing or flexible clinical protocols.

To address these challenges, simpler and more accessible DSLs without and/or less formal underpinning like DiaFlux and Conceptual Graph Formalism offer heuristic-based or graphical approaches that prioritise usability over formal verification. DiaFlux emphasises simplicity and visual representation, making it suitable for protocols like sepsis treatment [31]. Conceptual Graph Formalism uses graph-based logical reasoning for knowledge representation but still demands a formal computing background [32]. Both approaches, however, lack robust editing environments that balance ease of use with formal correctness. This gap can be effectively addressed by projectional DSLs, as discussed in the following sections.

In veterinary medicine, the situation is not quite so advanced in the development and uptake of computerised clinical protocols and corresponding DSLs. According to our knowledge, there is no DSL developed to describe veterinary clinical protocols. Of course, it is possible that DSLs were used in the development of other veterinary information systems such as CDSSs but no DSL has been published in academic literature or became open source.

Although no DSLs are explicitly designed for veterinary protocols, the integration of computerised decision support systems exemplifies a potential groundwork for their development. For instance, systems like the one for managing canine idiopathic epilepsy utilise patient data to provide clinical guidance, mimicking the structured approach of DSLs, yet without formal designation as such [33]. Additionally, recursive partitioning models that help predict outcomes in horses with abdominal pain demonstrate an approach to data-driven decision-making protocols, hinting at the foundational elements necessary for future DSL formalisation [34]. The latest Feline Diabetes application [35] launched by the Royal Veterinary College is a solution to assist owners/vets with handling pet diseases remotely. The latter justifies the importance of applying CPGs inside an information system to facilitate diagnosis and drive decision-making. Advancements in radiographic protocols and the incorporation of clinical skills laboratories and online collaboration tools in veterinary education underscore a movement towards systematic protocol optimisation. The information systems can also be a source of knowledge, a tool for skills self-assessment for vets, and could enhance good practice. Assisting to stage a cat with CKD or supporting the vet's confidence to diagnose a cat in stage I, through a veterinary Information System is important as this could help the clinician to make decisions. Adding a third tool to confirm staging will lead to a significant improvement in pet care, and pet well-being, and could improve the overall survival of the pets. These advancements suggest that DSLs could play a critical role in enhancing learning and practice by standardising procedures [36,37]. Ethical and practical considerations, such as the need for informed consent and ethical oversight in veterinary clinical research, further highlight the need for standardised approaches that could be streamlined by DSLs [38]. While the literature has not yet formally recognised DSLs in veterinary clinical protocols, the ongoing integration of decision support systems and protocol optimisation suggests a trajectory towards such developments, potentially revolutionising standardisation and efficacy in veterinary practices. Additionally, there is some work on the adoption of CDSS for veterinary applications as described in [33], and [39]. None of these works targets computerised clinical protocols.

Funding in veterinary research has always been more scarce than in human medicine due to prioritisation. However, there is great potential in applying lessons learned from human medicine to veterinary medicine, always considering that veterinary practices can be a substantially different problem. For example, there is limited access to referral cases in veterinary medicine making hospital tests more difficult [40] than in human medicine.

2.3. Why MPS JetBrains projectional DSL development as a solution?

Starting from the research in human medicine and DSLs, we initially investigated utilising a DSL for human medicine clinical protocols. However, we decided to develop a new DSL for veterinary clinical protocols for the following reasons:

1. The veterinary domain has different requirements than the medical domain especially when considered at the system level. For example, the patients in human medicine are the main actors where we extract information whereas the pet information is provided by pet owners and veterinarians as the pets themselves cannot create them.
2. There are problems in each category of the DSLs for human clinical protocols such as difficulties to use in the ones with formal underpinning and not enough rigour in the ones with no formal underpinning.
3. The re-development of the DSL would involve directly veterinary professionals leading to increased chances of adoption taking into account that adoption is a major problem in human medicine. Specifically, the most prevalent impediments to adoption were as follows: smooth integration with existing systems [41,42]; requirement for clinicians' training [43]; not enough engagement by medical professionals [33]. Therefore, we decided to develop a DSL for veterinary clinical protocols and specifically for CKD for cats as it is one of the most common diseases that veterinary practices face.

To address these issues and mainly the ones raised in item 2, we adopted the projectional editing approach by MPS JetBrains [44]. Specifically, it enhances the creation of clinical guidelines by integrating robust features such as real-time syntax and semantic checks, automated refactoring, and a user-friendly integrated development environment that supports both textual and graphical elements [45]. This advanced environment not only improves usability for domain specialists by simplifying the interaction with complex formal semantics but also ensures the consistency and correctness of the guidelines through formal semantic enforcement and the easy integration of new language constructs and validations [46]. As a result, projectional DSLs developed with MPS can significantly improve the accuracy, reliability, and maintainability of clinical guidelines, thereby fostering better adoption and effectiveness in clinical settings.

More details on the DSL will be provided in the following sections.

3. Proposed solution – CKD4cats DSL

3.1. CASE STUDY – CKD for cats clinical protocol

The CKD for cats clinical protocol is a complicated clinical protocol that is very important and frequently required in veterinary settings. It consists of four stages that require close attention through testing and visits to veterinary Clinics. The management of these cases is the most complicated among the cases that a veterinary Practitioner has to face.

In this paper, we based our clinical protocol description on IRIS clinical protocol [47]. In this clinical protocol, the four stages and corresponding treatments are detailed.

3.2. CKD4cats DSL high-level description

Detailed descriptions of the structure, the usage and the code generation part of the DSL are presented in the following subsections.

Our work was initially inspired by the DSL in [48]. Our main difference with this work is the different and complex veterinary protocol that we developed while addressing the particular requirements of the veterinary domain. We enhanced the constraint elements for the domain-users to provide additional guidance according to the veterinary domain users' requirements.

For our DSL development, we opted for a textual DSL development environment—the projectional editor by JetBrains MPS rather than a graphical DSL alternative. This choice was driven by two main factors. First, from a technical perspective, textual DSLs offer numerous advantages to tool developers such as language composition and integration with existing software systems, as discussed in [49]. Second, it was the preferred option of the collaborating veterinarians for this project, as it more closely resembled the MS Word documentation used for clinical protocols and the form-like input they were accustomed to. Their feedback during the evaluation phase reinforced this original decision.

3.2.1. CKDStageTreatment4Cats DSL

Language structure (abstract syntax)

After a thorough analysis of the CKD clinical protocol for cats [47], we identified the main parts of the structure that a DSL for CKD management for cats should consist of and are detailed as follows:

1. Stages of the CKD protocol, four in this case,
2. Supported statements such as Apply-Treatment, etc,
3. Managing conditions such as dehydration, proteinuria,
4. Measurements such as blood pressure measurement, etc.
5. Evaluation which is used to create the rules of the CKD protocol such as the ranges of normal values for blood pressure measurements.

The correspondence between the CKD protocol document and the high-level architecture parts of the DSL is depicted in Fig. 1 and was extracted by analysing the document in collaboration with the veterinary professionals. This figure gives the "parsing" method of the clinical protocol document that took place within meetings and led to the design and development of the DSL. This analysis could not

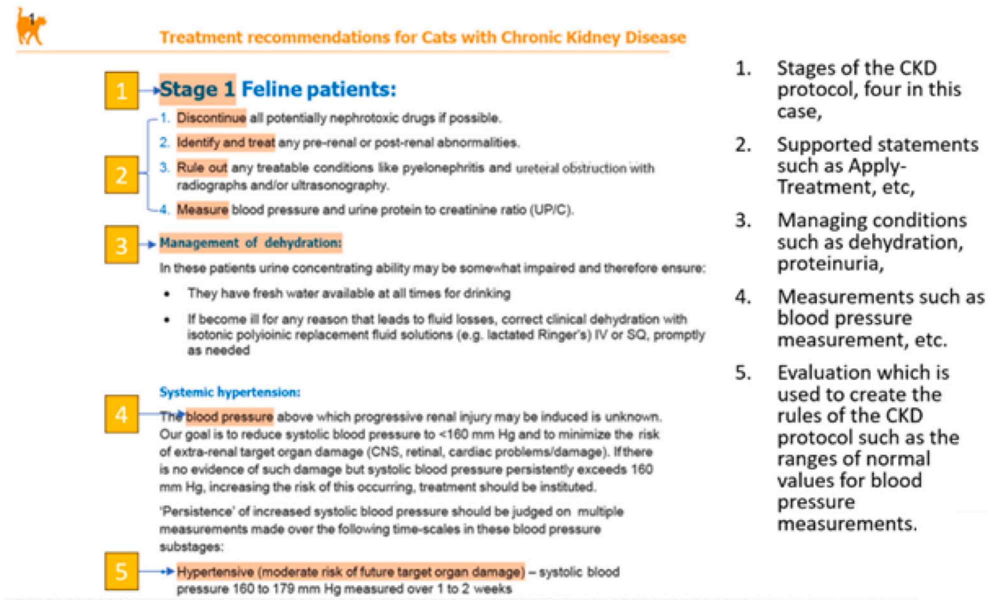


Fig. 1. CKD for cats clinical protocol document with architecture parts correspondence.

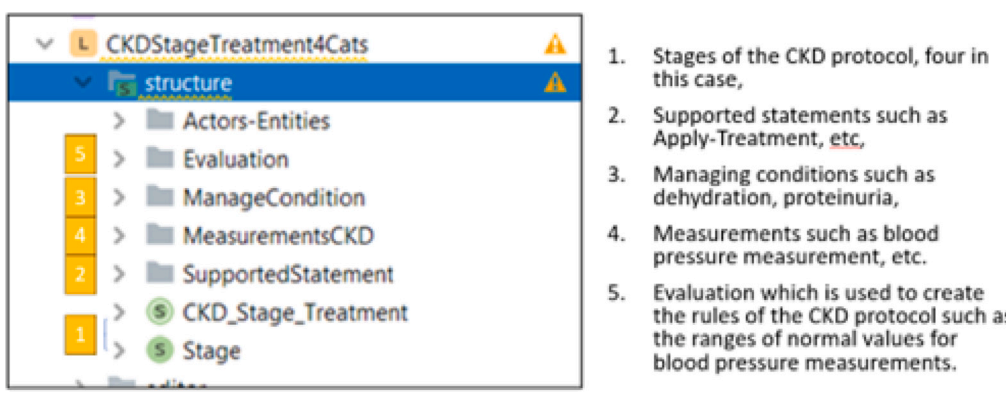


Fig. 2. MPS folder structure for CKD for cats clinical protocol high-level architecture.

have happened effectively if technical and non-technical users had not convened in one place.

In Fig. 2, these parts are depicted as they were defined as folders in the MPS structure. Please note that the Actors-Entities folder exists in our DSL but is not used by the current code. It was added for the future extensibility of the DSL with access control mechanisms. This would contain different user interfaces for technical and non-technical users and was initially of interest but deemed outside the context of the current development.

Each of these parts will be detailed as follows:

1. Stages of the CKD protocol.

To give an example of how the DSL concepts are implemented, we will discuss the code for the first two concepts: the *Stage* and the*CKD_Stage_Treatment*.

The *CKD_Stage_Treatment* consists of *n* stages as depicted in Fig. 3. Please note that to make the DSL extensible to other clinical protocols, the multiplicity of the stages is *n* instead of *4* which is the total number of stages of the CKD for cats clinical protocol as defined in [47]. Other clinical protocols might require a different number of stages.

In Fig. 4, it is also depicted that there are two types of statements that need to be supported for the specific clinical protocol:

1. General statements with the concept *Supported_Statement*,
2. Managing conditions with the concept *Manage_Condition*,

On the left-hand side of the above figures, we can see the logical view of the DSL and on the right side the corresponding concept.

2. Supported statements such as Apply_Treatment, etc.

The general statements that have to be supported are as follows: *Apply_Treatment, Discontinue_Drugs, Disease_Resolved, Dose_Reduction,* etc. These are usually instructions to the veterinary professional on an action to be taken. More comments and information on this will be added in the future Web version of the code so that a new veterinary professional can understand the reasoning and the course of actions to follow. For example, if the statement *Discontinue_Drugs* is chosen, then a list of drugs the Vet needs to discontinue will be given for the case of CKD. That can be very informative for young veterinary professionals as they would probably have had to look it up on the internet or other sources before making the final decision. This is the type of information/task [50] that requires memorising and an IT system can contribute and get easily accepted by medical professionals.

3. Managing conditions such as dehydration, proteinuria.

The second important clinical protocol structure that needs to be supported is the *Manage_Condition*. Inside managing a condition, all the regular supported statement can be added as before, and measurements can be ordered and evaluated. Inside the CKD for cats protocol, there are cases where the veterinary professional would need to manage other conditions such as Proteinuria, Systemic Hypertension, Dehydration, etc.

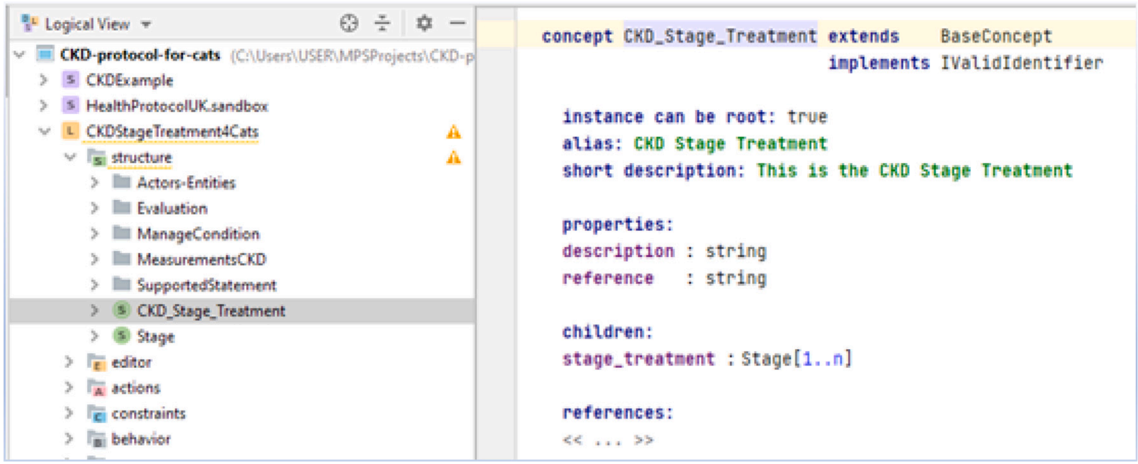


Fig. 3. *CKD_Stage_Treatment* concept.

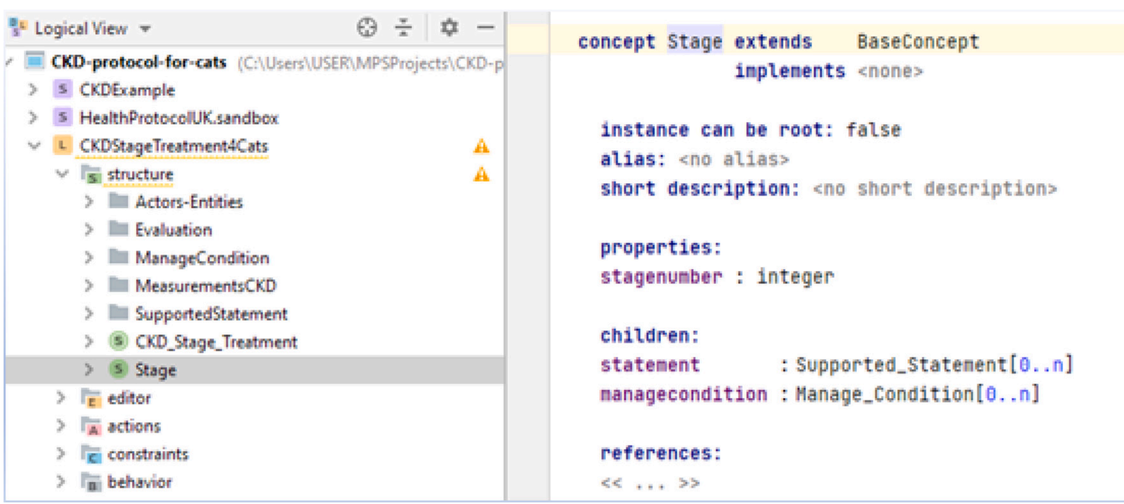


Fig. 4. *CKD_Stage_Treatment* concept.

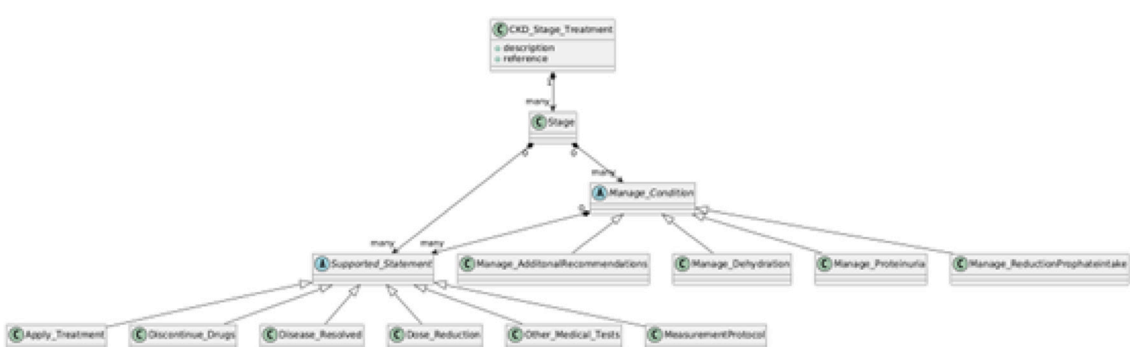


Fig. 5. DSL metamodel for Stages, Supported statements and Managing conditions.

To summarise the above points and enhance clarity and visualisation, Fig. 5 presents an overarching high-level metamodel diagram of the DSL concepts. The diagram illustrates how the *CKD_Stage_Treatment* is composed of multiple *Stages*, with each *Stage* encompassing *Supported_Statements* and *Manage_Conditions*.

Additionally, for the remaining points, we will not include MPS screenshots of their corresponding implementations, as they are straightforward and follow the same approach outlined for the first three parts of the structure: *(1) Stages of the CKD protocol, (2) Supported statements such as Apply_Treatment, and (3) Managing conditions such as dehydration and proteinuria.*

4. Measurements such as blood pressure measurement, etc.

The fourth important clinical protocol structure that needs to be supported is the *MeasurementsCKD*. Inside the CKD for cats protocol, there are cases where the veterinary professional would need to take several measurements (e.g. ProsphateConcentrationMeasurement, UPCProteinuriaMeasurement, etc.) to assist in diagnosis and decision-making.

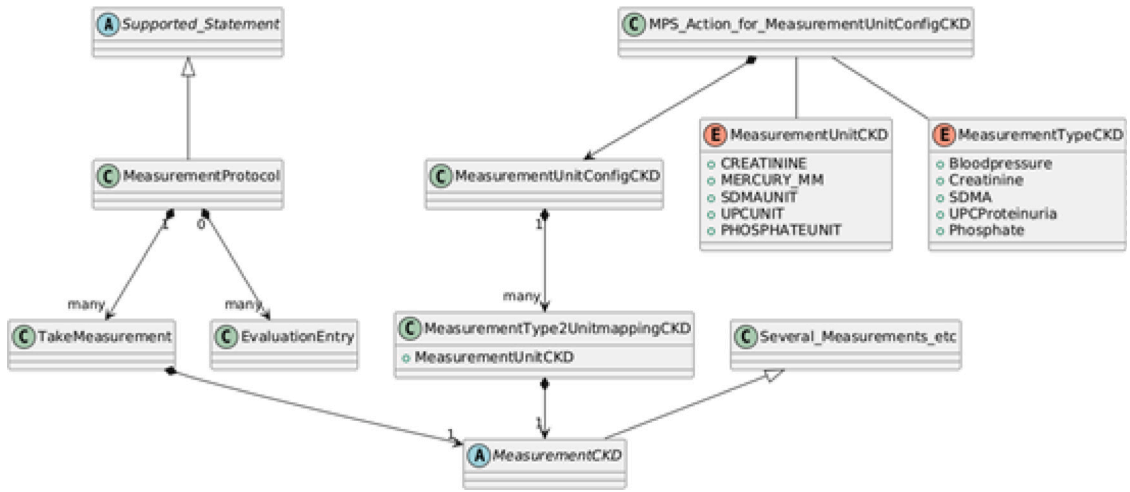


Fig. 6. DSL metamodel for Measurements part.

As illustrated in Fig. 6, the measurement component of the DSL can be described in detail as follows: The *MeasurementUnitConfigCKD* concept was introduced to establish the correct association between measurement units (e.g., mg/dL from the *MeasurementUnitCKD* enum) and measurement types (e.g., Creatinine from the*MeasurementTypeCKD* enum). This process is initiated by an MPS action (*MPS_Action_for_ MeasurementUnitConfigCKD*) when the M*easurementUnitConfigCKD* concept is instantiated. Subsequently, the *MeasurementType2Unit- MappingCKD* concept, facilitated through MPS editor substitute/ transformation menus (*MeasurementOperandAdapter*), ensures that the appropriate types are suggested and utilised by the domain user.

5. Evaluation is used to create the rules of the CKD protocol such as the ranges of normal values for measurements.

The fifth important clinical protocol structure that needs to be supported is the *Evaluation* part. In this part, the ranges are defined for the corresponding measurements and the checking of correct units associated with them takes place. To perform these tasks, we utilise MPS structures such as MPS concepts, enumerations and actions.

As illustrated in Fig. 7, the evaluation component of the DSL is detailed as follows: Within this part of the DSL, we define the measurement ranges (represented by the *MeasurementRange* concept) and their corresponding outputs (captured by the *OutputResult* concept). The MeasurementRange concept accommodates both unary operators, such as <=, <, >=, and > (modelled by the*MeasurementUnaryOperator* concept), as well as binary operators (represented by the *MeasurementBinaryOperator* concept). The units and types for these measurements were covered in the previous section. The*OutputResult* is assigned to trigger actions, such as *AddReminderAction*, based on the clinical protocol specifications.

On reflection of the overall DSL design, we used simple structures to design this DSL such as inheritance for the different types of measurements, conditions, enumerations for the units, etc. The choice was for ease of development and code generation. However, we plan for a more design-for-extensibility version. This will contain more abstract concepts and the rest will be extensions.

For a comprehensive reference of all elements in the DSL, please refer to the publicly available source code on GitHub [51]. Additionally, a complete overview of the metamodel can be found in Appendix A.

Please note that all the information provided in this subsection (***Language structure (abstract syntax)***) of Section 3.2.1 pertains to the abstract syntax of the DSL. This includes the screenshots from MPS in Figs. 2–4 which illustrate how the Abstract Syntax Tree (AST) is defined in MPS. These figures do not represent the concrete syntax of the DSL, which is described in detail in the following section.

Language usage (concrete syntax)

In this subsection, the language usage will be presented. The target users of this DSL are veterinary professionals who are interested in developing computerised clinical protocols.

Language usage is seamlessly integrated into the MPS JetBrains environment through the MPS solutions space. This setup enables both the language developer and the domain expert to test and refine the DSL collaboratively within the same environment during the DSL development (co-creation) phase. Typically, end users do not have access to the development environment where their tools are created, and veterinarians are no exception.

Specifically, the DSL development process relies on co-creation and frequent collaboration between the language developer and the domain expert. A unified development environment, such as the MPS JetBrains platform, significantly facilitates this iterative process. While this setup offers clear technological advantages during development, it is not practical for domain users at the deployment stage. Instead, a simplified and user-friendly environment will be provided to meet their needs during DSL deployment.

Fig. 8 illustrates the core interface presented to the domain-user in both cases (development and deployment), demonstrating the use of the DSL to describe the CKD protocol for cats. In this example, all the key components discussed in the Language Structure section such as supported statements, managing conditions, ordering, and evaluating measurements are included. This is what the domain-user (veterinary professional) would input to create a protocol. Different colours have been used to enhance readability and improve the user experience.

It is important to note that the main elements of the DSL are easily identifiable in the final DSL interface and are also depicted in Fig. 8 with the corresponding numbers defined in Fig. 1.

Projectional Editing: User Experience for Non-Technical Veterinary Professionals

The requirements posed by non-technical domain users are fundamentally different than the requirements posed by programmers/ language engineers [52]. According to [52], domain-users prefer simple editors and to be accessed via the web avoiding local installation. In [53], it is mentioned that projectional editors are easier to learn for non-technical users than programmers that are more accustomed to text editors. The reason behind it is most likely the fact that programmers are used to “programming-style” editing and the textual DSLs are out of the ordinary for them. It is also stated in [53], that this hypothesis would require a well-designed human subject study that was beyond the scope of this manuscript. This remark is inline with the general recognition by [54] that the domain-specific language research lacks experiential research.

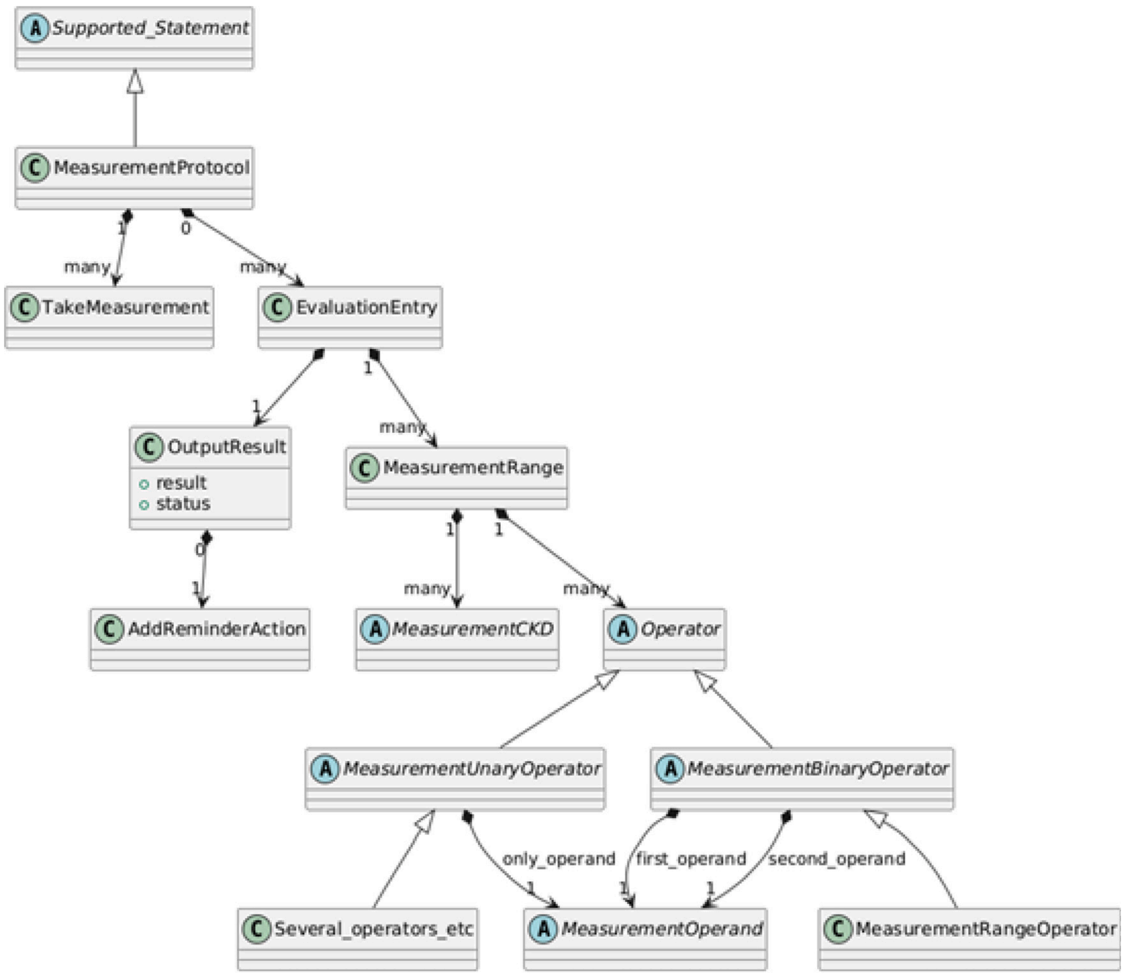


Fig. 7. DSL metamodel for Evaluation part.

In this work, the target users were veterinary professionals who are considered non-technical domain users and we tested it in a moderate number of 20 that was feasible within the veterinary company. More details are provided in the evaluation section.

The benefits offered to the domain-user (veterinary professional) belong to the general benefits that are being offered by projectional editors and can be summarised as follows:

1. At every step, with *Ctrl+Space*, the veterinary professional can see all and only the available options,
2. For the Evaluate-check step, the veterinary professional can see only the measurements that were ordered.
3. Guidance through the correct in-context drop-down menus was offered to assist them in the process
4. Constraints that were implemented under the hood provided a "correct-by-construction" clinical protocol.

These elements proved to be invaluable from the perspective of veterinary professionals, as detailed in the Evaluation section.

Error checking

Error checking is also illustrated in Fig. 8, where two red underlines indicate errors in the MPS environment. The first underline highlights a unit error, where the user incorrectly input *mmHg* as the unit for measuring *Blood Creatinine Concentration* instead of the correct unit, *mg/dL*. The correct unit is shown in the other lines, where no errors are flagged. The second underline points to an error in the measurement range values, where the first value on the left side of the range operator ("-") is greater than the second value. Both of these error-checking mechanisms were highly rated by users, as they effectively guided them in creating accurate clinical protocols. Also, this figure depicts a dropdown menu with available options, illustrating an example of the limited selection accessible to the domain-user.

The implementation of these error-checking mechanisms was based on a combination of MPS *concepts* and *actions*. In Fig. 9 the *check_MeasurementOperandAdapter* action uses the *MeasurementUnitConfigCKD* concept to search if the unit-measurement pair inserted exists in the predefined configuration file that contains the correct unit-measurement pairs. If it does not match one of the existing options, an error is raised *"unit x for type y not allowed"*. The configuration file is constructed if it does not exist in the beginning (when the user starts the protocol development) and its content is depicted on the right side of Fig. 9.

It is worth noting that, while the error-checking mechanism received high praise from domain users, we plan to enhance it further in future iterations. This includes offering fix suggestions and displaying error details through tooltips when hovering over incorrect text.

MPS JetBrains Generator

From the above DSL, we used the MPS code generation mechanism to produce Java code that can consumed by other IT environments used in the veterinary practices.

An example of the code that had to be written is presented in Fig. 10.

In this Fig. 10, the corresponding inspector window depicts the full definition of macros.

According to our experience over several projects, the MPS JetBrains generator is an excellent generator approach as with carefully

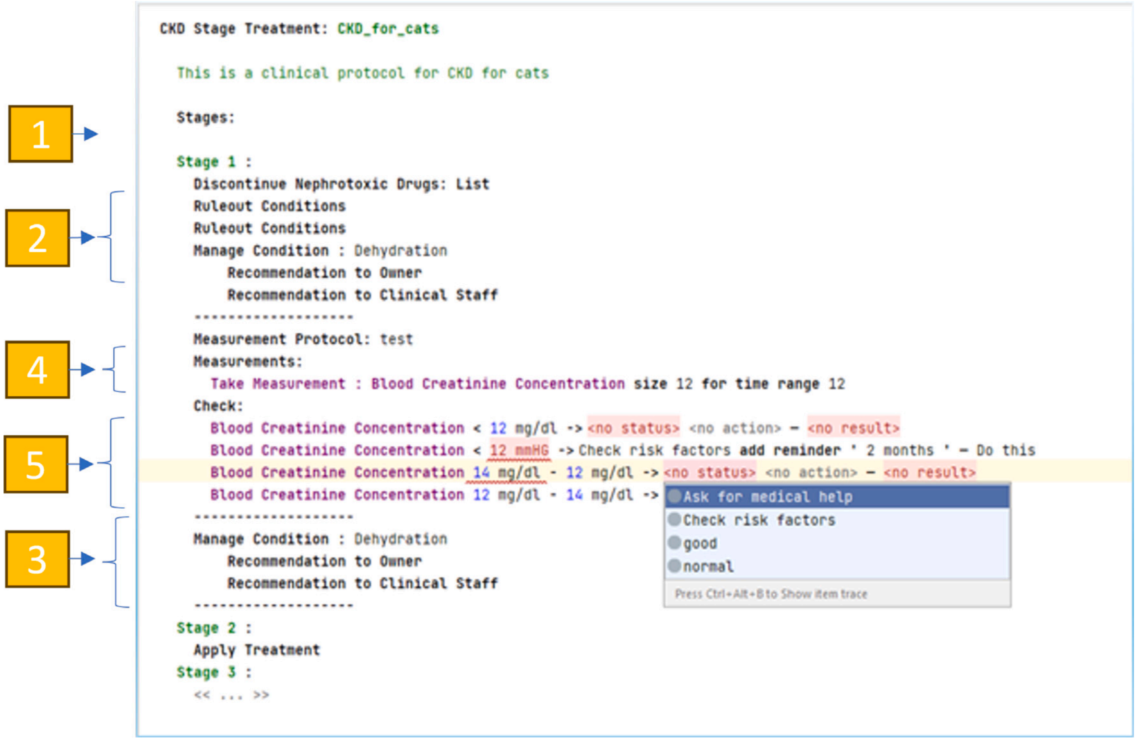


Fig. 8. Final DSL interface for veterinarians for CKD for cats protocol.

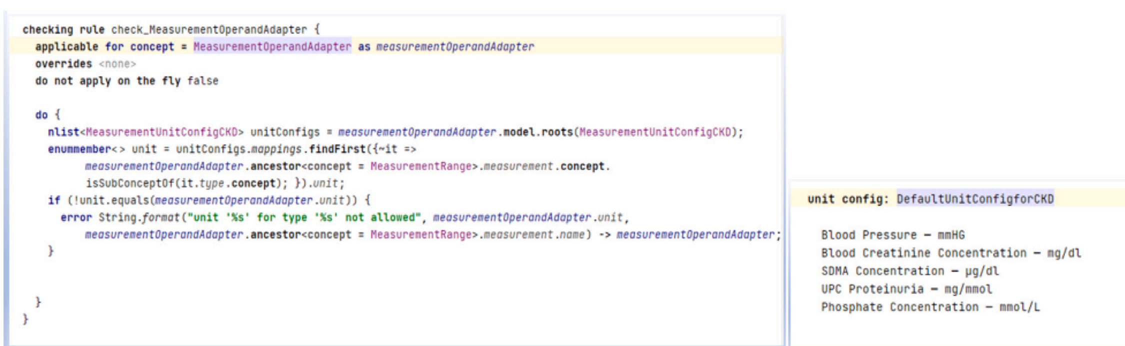


Fig. 9. MPS *action* code for units error checking.

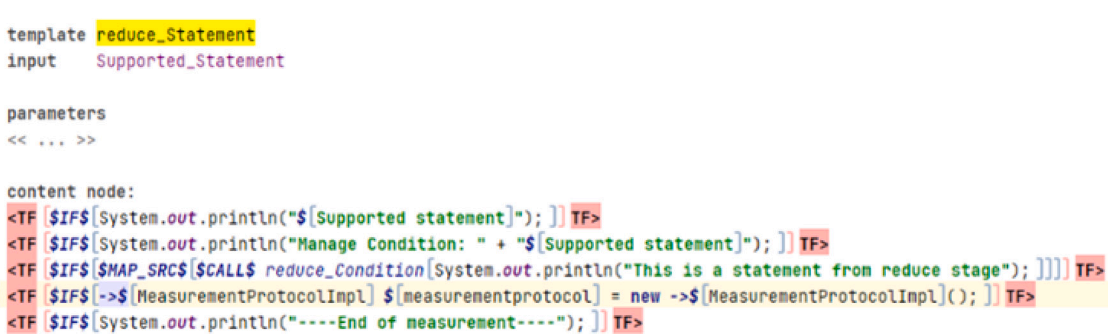


Fig. 10. The Generator structure for statements.

provided templates and a few lines of code, you can create robust systems. More on our experience evaluation of the generator is provided in Section 4.3.

3.3. CKD4Cats DSL usage (generated code) within other veterinary information systems

The CKD clinical protocol will be developed by veterinarians to establish the "correct" computerised clinical protocol. This protocol, implemented as code, can then be integrated into various veterinary information systems, such as web-based platforms for clinical decision support. By embedding a clinically validated protocol at the core of these systems, developed through the DSL, we ensure accuracy and reliability in decision-making.

The code generated by the DSL is designed for seamless integration with other information systems that may manage data about individual cats. Notably, the DSL itself is not tailored to individual cats but instead generates general-purpose code aligned with the clinical protocol, co-developed with veterinary professionals to ensure accuracy and validity. Databases containing specific information about individual cats can be utilised when the DSL-generated code is incorporated into these systems. These combinations will enable answering questions like, "*What course of action should be followed for a cat with Stage 2 CKD?*". This is an initial interpretation of the generated from the DSL usage as has been described in [48].

4. Evaluation

We evaluated the CKD4Cats DSL according to the Quality characteristics defined in the paper [55], as well as feedback from CVS veterinary professionals and language developers. Our study was designed as a structured or exploratory evaluation [56] tailored to the domain-specific context of veterinary medicine. We aimed to ensure a systematic and unbiased assessment of CKD4Cats through structured workshops, demonstrations, and participant feedback.

Please note that our evaluation involved a reasonable sample of participants which often comes with the territory of DSLs' evaluations, representing an initial step towards broader adoption. However, the scope was limited due to time and resource constraints.

4.1. Experiment setup

To evaluate the CKD4Cats DSL from both veterinary and language engineering perspectives, we designed an evaluation experiment and developed a corresponding questionnaire. We followed a methodology that we developed through our experience with DSLs and refined it with a workshop approach.

We evaluated the developed DSL from two primary perspectives: non-technical veterinary users and language engineers. In this context, " language engineers" refers specifically to DSL developers, not general software developers. Consequently, the target population for the evaluation comprised non-technical veterinary users to assess domain-specific usability, and language engineers to evaluate the technical aspects of the DSL.

For the evaluation, we initially followed the quality characteristics defined in [55], which we have applied in all our previous DSL studies [57–59]. However, based on insights gained from these earlier studies, we refined our approach to improve the evaluation process. Our updated method consisted of three key elements: a demo presentation by the DSL developer, followed by a hands-on workshop/training session for veterinary professionals, and concluding with a questionnaire to collect feedback. This structured method was chosen to enhance the validity of the evaluation by ensuring consistent participant exposure and interaction with the DSL, while also improving the quality of the feedback to guide future DSL improvements.

We did not include the language and execution dimensions, as defined in [55], which address the development effort required for the DSL. This aspect requires larger-scale, longitudinal research beyond the scope of this study. However, it is worth noting that particularly in our project, development effort was minimal due to the prior expertise of the language engineer, despite MPS's typical learning curve. The DSL's first version was co-developed in about a month, with two more months for iterative improvements. The first author, with five years of experience in DSL development, significantly expedited our project's timeline. However, for new researchers, especially those with prior experience in model-driven engineering, we estimate that developing a similar DSL would require approximately six to eight months. Without such modelling experience, the time investment would likely be higher.

Step 1: Demo Presentation
The experiment began with a demo presentation delivered by the DSL developer. This presentation, consisting of ten slides and lasting less than ten minutes, introduced the primary logic and usage mechanisms of the DSL. It also outlined the tasks that the veterinary professionals would complete during the hands-on session. The presentation was standardised across all participants to ensure consistency in the instructions provided.

Step 2: Hands-On Workshop
In the second phase, participants engaged in a controlled hands-on workshop where they applied the knowledge from the presentation to create a clinical protocol using the DSL. Non-technical veterinary users were asked to follow the steps outlined in the demo. The language developer observed and recorded their interactions with the DSL, following a simplified version of Krug's usability testing method [60]. Key usability metrics, such as ease of learning and task completion, were recorded.

To ensure comparability, all participants received the same instructions, were given access to the same DSL editor, and had a similar timeframe for completing the task. The learning curve was measured by the number of errors made and the time taken for users to achieve proficiency, with most users demonstrating competence after a few mistakes. The overall timeframe for learnability was set to half a day, which provided a suitable benchmark for evaluating the DSL's accessibility for non-technical users.

Step 3: Questionnaire Feedback
Following the hands-on session, participants were asked to complete a standardised questionnaire designed to gather quantitative and qualitative feedback (see Appendix B). The questionnaire was distributed electronically, with a link provided at the end of the workshop. It included quantitative items using a Likert scale ranging from 1 (low) to 5 (high). In previous studies, we used a 0–10 scale, but for this evaluation, we switched to the Likert scale due to its proven efficiency in capturing participant responses more effectively.

Additionally, the number of questions was reduced to seven, a significant decrease from our earlier research. This change aimed to reduce participant fatigue and improve focus by asking concise and relevant questions. We also improved the quality of the questions based on research by [61], which highlights five common pitfalls in Likert-style questionnaires: ambiguous, awkward, biased or leading wording, conjunctions, and double negatives. Initially, our questionnaire contained some of these issues. For example, the original question, "*Do you think it was easy to follow the presentation?*", was revised to "*PRESENTATION: How would you rate your ability to follow the presentation?*" to reduce ambiguity and bias.

Open-ended questions were also incorporated to allow participants to share additional insights freely. These questions, such as "*Are there any other mechanisms you would like to see supported that are currently missing?*" and "*Where do you think this could be used in a veterinary practice?*", were designed to elicit domain-specific feedback relevant to the evaluation goals.

4.2. Results

A total of 20 participants contributed to the evaluation, comprising 16 veterinary professionals and 4 DSL language developers. The veterinary participants were partly mid-career experienced professionals and partly new veterinarians at a rate: of 4 out of 16 (25%) mid-career, and 12 out of 16 (75%) were new professionals. They all had computing skills and experience from the basic use level (simple use of software packages) to advanced (advanced use and some had basic programming skills) at a rate: 12 out of 16 (75%) had basic computing skills and 4 out of 16 (25%) were advanced. Other than these, no other characteristics of the participants were important for the study. The language developers were all at advanced level and mid-career individuals due to the advanced level of language engineering area of work. Needless to say, none of the developers of this DSL which are the authors of this paper or the veterinarian that helped in the development participated in the evaluation phase. Note that language engineers were included in the evaluation to assess aspects such as maintainability, which non-technical users are unable to evaluate. The same questionnaire was used for all participants. In future work, different evaluation cycles will be conducted with larger cohorts to enhance the robustness of the results.

Please note that in this evaluation, the same questionnaire was administered to all participants; however, the language engineers did not answer Q6 and Q7. They responded to all other questions, as they were trained during the workshop to understand the basics of the clinical protocol. This approach represents a limitation in our current work, arising from the need to gather feedback on all quality characteristics of the DSL through a single questionnaire. We plan to address this with separate workshops and questionnaires for the different user groups.

In addition to the above evaluation-conclusions that were based on qualitative analysis of the questionnaire responses, we performed quantification of all the responses and provided a chart representation for the relevant metrics as depicted in Fig. 11.

Most of the quantification results came directly from the likert scale questions. This applies for the questions Q2-Q5 of the questionnaire (for the corresponding questions see Appendix B). The question Q1 was mostly informative to assist interpretation of the results. The open-ended questions Q6, Q7 were used to draw the text conclusions and quantified results for *Expresiveness* and *Functional suitability*. For the quantification of the qualitative results we chose to use the coding open-ended questions method [62] for its simplicity and applicability to our requirements. For future work with larger cohorts, another method including text analysis tools such as NVIVO [63] will be utilised.

4.2.1. Veterinary professional (non-technical domain user) results

The results from the veterinary participants are presented as follows:

Expressiveness: The language is very expressive for the purposes of the specific protocol. In future work, it will be extended to more clinical protocols. There were some comments from the veterinary professionals that they would like to have *"the option to add clarifying text at points"* which limited the score for expressiveness. They also noted things such as *"to be able to have different options for specific measurement types"*.

Maintainability: It scored low for maintainability as it would require language engineering background and the domain users would not be able to maintain it.

Usability: Regarding usability, the language scored high as it takes a couple of trial and error cycles to get used to the different features. This was surprisingly positive and we believed it stemmed from the number of features being small and easy to learn. It took around half-a-day for the targeted learning process for training the domain-users which is a relatively good result. These results were partly concluded from the veterinary professionals such as: *"It was hard to start with but once I started, very effective"*.

Reliability: The language was very reliable as it did not allow any errors. For example, the DSL automatically introduced measuring units and "corrected" all wrong usage.

Productivity: The use of this language enabled a quick turn-around time for the development of a clinical protocol. These conclusions were primarily derived from the questionnaire responses and observations made during the workshop. *T* hese results were partly concluded from the veterinary professionals such as: *"It was hard to start with but once I started, very effective"*.

Functional suitability: The language scored very high according to its suitability for all the functionality required by veterinary professionals. The data validation was an impressive characteristic, and more validation options were suggested such as restricting users with only permitted options in all parts of the interface. These results were partly concluded from the veterinary professionals such as: *"it covered all my needs"*, *"No other mechanisms to be supported"*, *"Had all required"*.

The scoring for all individual elements is presented in Fig. 11.

4.2.2. Language engineer results

The results from the language engineers participants are presented as follows:

Expressiveness: The language was rated highly for expressiveness by the language engineers. However, this evaluation was based on their limited exposure during the initial training session, and the result should be interpreted with caution as it may not accurately reflect their overall experience.

Maintainability: It scored low for maintainability as it would require specialised effort. The language engineers were asked to grade this objectively from the perspective of a non-technical user.

Usability: Regarding usability, the language received high scores from the language engineers. However, this result should be interpreted with caution, as language engineers are accustomed to working with such interfaces. In future work, we plan to refine this question to explicitly compare this DSL with other DSLs based on their professional experience.

Reliability: The language engineers marked the language as highly reliable as there were a number of input-checking. This stemmed from their experience with other languages and not from the requirements perspective of the veterinarians.

Productivity: The language engineers scored the language slightly lower than the veterinarians on this aspect, which is unexpected given their prior experience with language engineering and the productivity it should afford them. We attribute this result to an error in how we interpreted the qualitative data, as the language engineers did not provide responses to Q6 and Q7. Separating the experiment design in future iterations will help address and mitigate these issues.

Functional suitability: The language also scored high by the language developers as "it provided the necessary structure" that enables the development of DSLs for CKD for cats clinical protocol.

The scoring for all individual elements is presented in Fig. 11.

It is worth noting that our work balances reliability and extensibility in DSL design, with iterations between language engineers and veterinary professionals ensuring the CKD clinical protocol was fully covered. Prioritising reliability for this specific application, we opted for a simpler design. Future versions will focus on extensibility, even if reliability is partially sacrificed, to enable domain users to maintain the DSL independently.

It is anticipated that this method will be applicable in more veterinary and other clinical protocols and will be integrated with existing information systems.

4.3. Contributions-discussion

The main contributions of this DSL can be summarised from two different perspectives: 1. Benefits for developing clinical protocols from the veterinary professional; 2. Benefits for the software developer that needs to integrate clinical protocols.

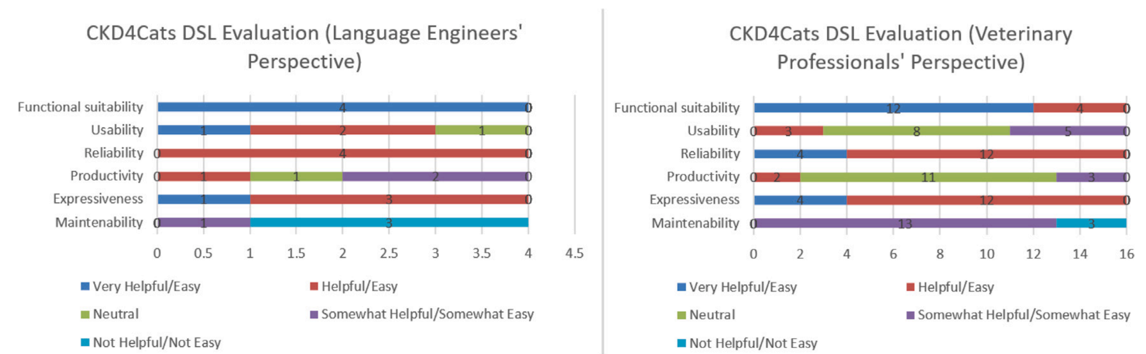


Fig. 11. Evaluation by veterinary professionals and language engineers.

4.3.1. Veterinary professional (non-technical domain user) perspective

The veterinary professionals who used the DSL expressed a generally positive impression of its usability and functionality. As non-technical domain users, they appreciated the intuitive nature of the projectional editor, which aligned well with their needs and preferences for simplicity and ease of use. Features such as context-sensitive suggestions available through *Ctrl+Space*, guided dropdown menus, and the ability to see only relevant options at each step significantly enhanced their interaction with the tool. The "correct-by-construction" approach ensured by underlying constraints was particularly valued, as it minimised errors and streamlined the creation of clinical protocols. These elements collectively provided a user-friendly experience, allowing the veterinary professionals to focus on their domain tasks without the complexity typically associated with programming environments. The evaluation revealed that these benefits were instrumental in making the DSL accessible and effective for veterinary professionals.

All in all, the perspective and the requirements of the non-technical user were the primary focus of this work. Although DSLs are known for their benefits to non-technical users, pure experiments of this nature are rare due to difficulties bridging the gap between advanced technical knowledge and domain users.

4.3.2. Language engineer perspective

The main advantage of this development from the language engineer perspective is the code generation mechanism.

On reflection, using the MPS generator was quite hard but worth the effort. We achieved a great result of having automatically generated code that is in alignment with the high-level model of the DSL. This is very important for the adoption and maintenance of model-driven techniques and specifically addresses the issue of misalignment between models and code in MDE [64]. In this case, we attempt to take a middle-ground approach as was taken when developing the mbeddr platform [65]. For that purpose and outcome, the difficulties we encountered using the MPS generator were worth the effort. Although the total lines of code required to write the code generator using MPS is minimal as a number, the complexity and effort required to write them is substantial. This is in contradiction with previous papers [66] on the matter where they used the total lines of code as a criterion of the ease of use. Our experience has demonstrated that the effort is substantial despite the lower number of total lines. On reflection, the benefit of the approach lies more on the code-centric part that is always aligned with the higher-level model. More details on the generation process are outside the scope of this paper as the generated code and its usage within veterinary information systems is part of future work.

4.3.3. Discussion on evaluation, limitations and future plans

The evaluation of the DSL using key software quality metrics — usability, reliability, productivity, maintainability, and functional suitability — provided valuable insights into its strengths and areas for improvement. Reliability and functional suitability emerged as key strengths, with the DSL preventing errors through automated unit enforcement and offering robust functionality for clinical protocol development. Productivity was also positively rated, with a quick turnaround time despite initial learning challenges. However, maintainability and usability presented challenges. The DSL requires specialised effort for maintenance, which could be a limitation for non-technical users. Additionally, usability could be improved by addressing feedback such as adding clarifying text and allowing more flexible measurement options.

Specifically, there is a trade-off between reliability and extensibility - maintainability. Several iterations took place between the language engineer and the veterinary professionals to ensure that the CKD clinical protocol was fully covered. Future potential versions were also discussed. However, a trade-off exists between ensuring reliability through restrictions and guidance for users and achieving future extensibility.

If the DSL were designed primarily for extensibility, there was a risk it would become overly general, compromising reliability. Given that our target was specific and reliability of the protocol produced was a high priority, we opted for a simpler DSL design.

Although our DSL has been specifically designed for the CKD4Cats clinical protocol, our findings indicate that it has the potential for generalisation to other clinical applications. Through our comparative work with CKD in human medicine [20], a substantially different protocol, we identified core structural similarities between the two. These shared elements suggest that our approach is not limited to a single condition but can be abstracted to a higher level, allowing it to accommodate multiple clinical protocols while maintaining accuracy and reliability.

Building on these findings, our DSL overcomes the accessibility barriers of formal DSLs while maintaining precision, offering a balance between usability and correctness. Unlike heuristic-based approaches that lack validation, our projectional editing ensures structured, intuitive modelling without programming expertise. As the first dedicated DSL for veterinary clinical protocols, it enables standardisation and decision support, adapting lessons from human medicine while addressing veterinary constraints. This innovation bridges the gap between research and practice, advancing protocol automation across both fields.

Despite the valuable insights gained into the applicability of the DSL for CKD4Cats, several limitations must be acknowledged. First,

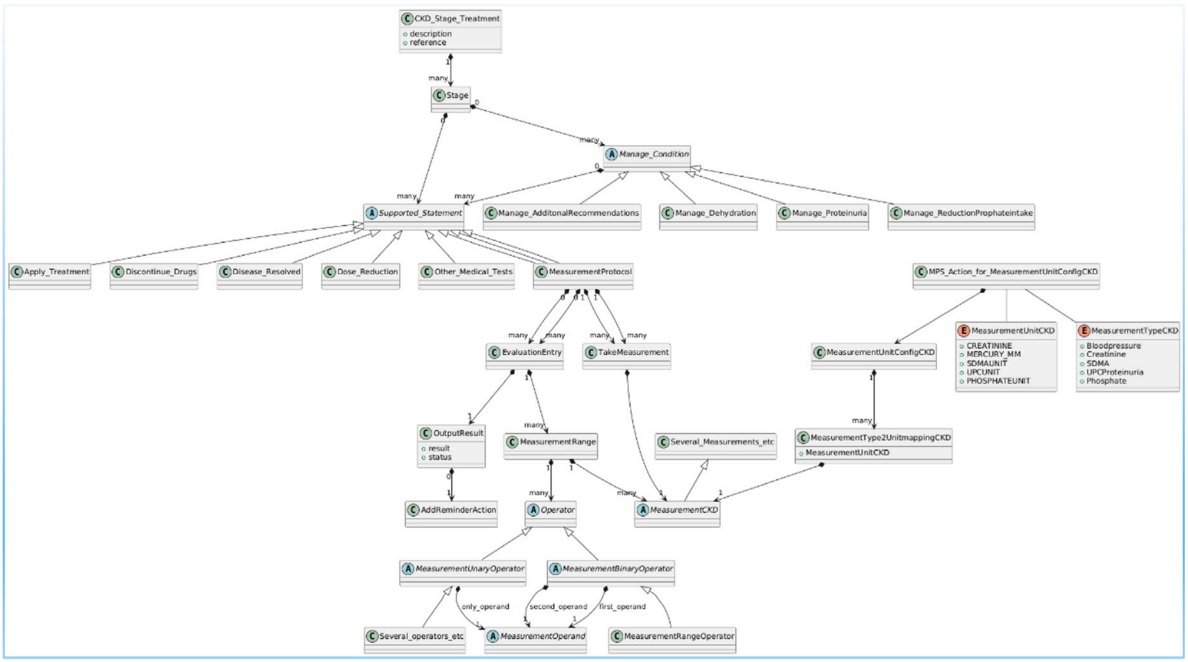


Fig. A.1. CKD4Cats DSL Metamodel.

the sample size of 20 participants is relatively small, which may limit the generalisability of our findings. Additionally, the evaluation was conducted in a controlled setting rather than in real-world veterinary practices, meaning external factors that could affect usability and adoption were not fully considered. Another limitation is the dependence on the MPS JetBrains environment, which may restrict the broader adoption of CKD4Cats unless further integrations with other platforms are explored. Finally, there is a potential learning bias, as participants received training before using the DSL, which could have influenced usability scores. Addressing these limitations in future work will help improve the transparency, credibility, and applicability of our approach.

Future iterations of our work will aim for a more general approach to DSLs for clinical protocols, prioritising extensibility even at the expense of reliability to accommodate future extensions. Ultimately, if the goal is to enable domain users to maintain the DSL independently, extensibility becomes a more critical factor. Our conclusions on productivity were drawn from questionnaire responses. Moving forward, we plan to extend this DSL, evaluate its effectiveness, and investigate potential causal relationships between metrics through controlled experiments.

5. Conclusions

In this paper, we proposed a domain-specific language (DSL) specifically tailored for the CKD4Cats clinical protocol, addressing the complexities of managing Chronic Kidney Disease (CKD) in cats — a very common and challenging condition in veterinary practices. We detailed the design and development of this DSL, motivated by the need for accurate, reliable, standardised information systems. The integration of computerised clinical protocols, essential in human medicine, has shown significant promise in improving clinical decision-making and care consistency. However, veterinary medicine has yet to fully embrace such advancements, particularly in the creation and utilisation of DSLs.

Our proposed DSL aims to bridge this gap by enabling non-technical veterinary professionals to develop computerised clinical protocols that ensure “correctness” through compliance with established guidelines. Initial evaluations of CKD4Cats have been promising, with veterinary professionals responding positively to the ease of use and the enforced accuracy provided by the MPS JetBrains projectional editor. This innovative approach not only enhances the management of CKD in cats but also contributes to the broader goal of standardising veterinary care practices.

The DSL evaluation highlighted strengths in reliability, functional suitability, and productivity, with quick turnaround times and robust clinical protocol development. However, maintainability and usability posed challenges, particularly for non-technical users. A trade-off emerged between reliability and extensibility, leading to a simpler DSL design prioritising reliability. Future iterations will aim to enhance extensibility, enabling easier maintenance by domain users. Our DSL, initially developed for the CKD4Cats clinical protocol, shows promise for broader application in other clinical contexts. Comparative analysis with human chronic kidney disease (CKD) protocols revealed significant structural similarities, suggesting our approach can be abstracted to accommodate multiple clinical protocols while maintaining accuracy and reliability. Our DSL enhances usability and precision in veterinary clinical protocols, enabling intuitive modelling without programming expertise, thus facilitating standardisation and decision support.

Despite valuable insights, our study’s limitations include a small sample size, controlled setting evaluations, reliance on the MPS JetBrains environment, and potential training biases. Addressing these in future research will enhance the DSL’s applicability and reliability.

6. Future work

Looking forward, our future work will include a more comprehensive usability evaluation, incorporating time metrics and expanding the sample size to align with established usability studies [67]. We also plan to refine our evaluation methods to further establish quality criteria for DSLs, following the approach of Challenger et al. [55] with additional refinements. Further interrelations between quality criteria will be explored through controlled experiments. Additionally, we will separate experiments between language engineers and domain users, addressing a current limitation in our study and obtaining more accurate measurements through controlled experiments.

To enhance maintainability, we will improve the interface with better error messages that domain users can easily understand, reducing the need for specialised expertise. Furthermore, we will follow advancements in the language workbench to leverage new features that support long-term maintainability and extensibility.

Additionally, we intend to integrate the developed DSL with veterinary information systems using web technology and test it in real veterinary environments. This will allow us to refine the protocol further based on more extensive case data.

Our long-term goal is to extend this work to include other clinical protocols in both veterinary and human medicine, exploring the potential for generalising the DSL to broader health-related domains. Through these efforts, we aim to establish a flexible and extensible framework that enhances clinical decision-making and protocol standardisation across multiple disciplines.

CRediT authorship contribution statement

Sofia Meacham: Writing – original draft, Validation, Software, Methodology, Investigation, Conceptualization. **Hessa Alfraihi:** Writing – review & editing, Validation, Methodology, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

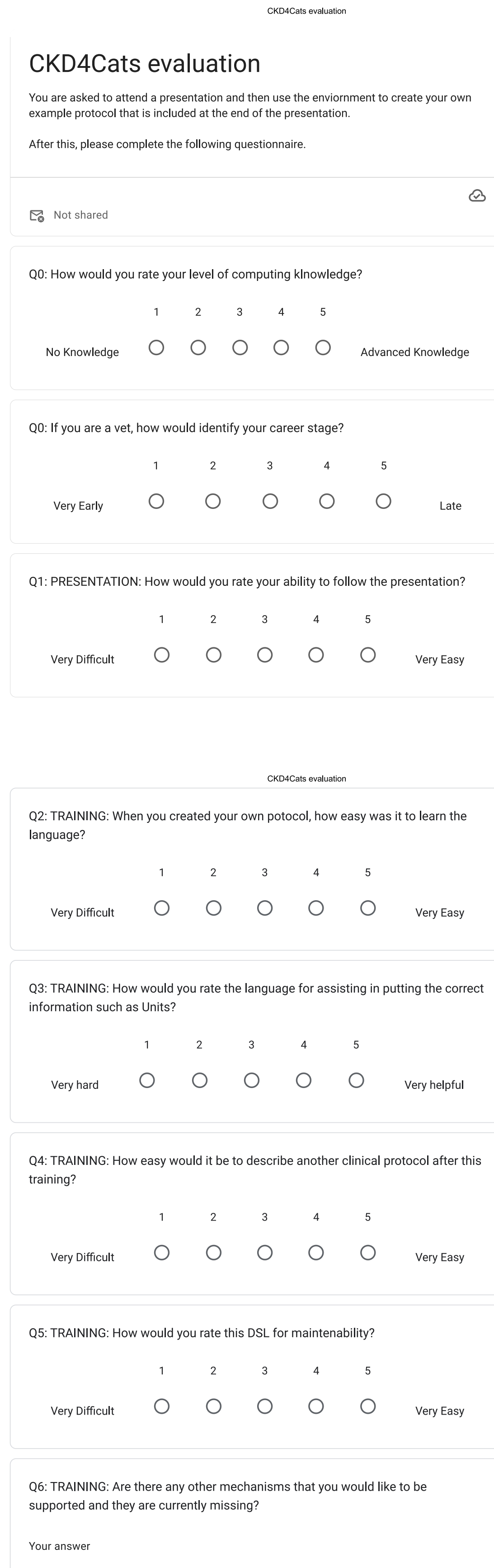
We would like to thank Michail Zavlaris and the CVS plc veterinary practice for their valuable time, and feedback that they provided for this project.

This research was funded by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2025R 411), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Appendix A. CKD4Cats DSL metamodel

See Fig. A.1.

Appendix B. CKD4Cats DSL evaluation questionnaire


CKD4Cats evaluation

CKD4Cats evaluation

You are asked to attend a presentation and then use the envionment to create your own example protocol that is included at the end of the presentation.

After this, please complete the following questionnaire.

Not shared

Q0: How would you rate your level of computing klnowledge?

No Knowledge 1 2 3 4 5 Advanced Knowledge

Q0: If you are a vet, how would identify your career stage?

Very Early 1 2 3 4 5 Late

Q1: PRESENTATION: How would you rate your ability to follow the presentation?

Very Difficult 1 2 3 4 5 Very Easy

CKD4Cats evaluation

Q2: TRAINING: When you created your own potocol, how easy was it to learn the language?

Very Difficult 1 2 3 4 5 Very Easy

Q3: TRAINING: How would you rate the language for assisting in putting the correct information such as Units?

Very hard 1 2 3 4 5 Very helpful

Q4: TRAINING: How easy would it be to describe another clinical protocol after this training?

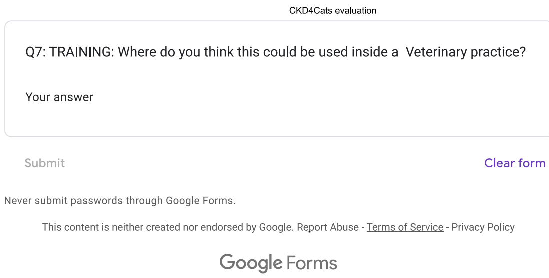
Very Difficult 1 2 3 4 5 Very Easy

Q5: TRAINING: How would you rate this DSL for maintenability?

Very Difficult 1 2 3 4 5 Very Easy

Q6: TRAINING: Are there any other mechanisms that you would like to be supported and they are currently missing?

Your answer


CKD4Cats evaluation

Q7: TRAINING: Where do you think this could be used inside a Veterinary practice?

Your answer

Submit Clear form

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. Report Abuse - Terms of Service - Privacy Policy

Google Forms

Data availability

The code is provided as open source on github: https://github.com/stsasakou/CKDprotocol4cats/tree/main.

References

[1] C.L. Marino, B.D.X. Lascelles, S.L. Vaden, M.E. Gruen, S.L. Marks, Prevalence and classification of chronic kidney disease in cats randomly selected from four age groups and in cats recruited for degenerative joint disease studies, J. Feline Med. Surg. 16 (6) (2014) 465–472.

[2] T. Kongtasai, Renal biomarkers in cats: A review of the current status in chronic kidney disease, J. Japanese Assoc. Veterinary Nephrol. Urol. 15 (1) (2024) 23–40.

[3] K.N. Lohr, M.J. Field, et al., Clinical Practice Guidelines: Directions for a New Program, vol. 90, National Academies Press, 1990.

[4] M. Pugliese, E. Voslarova, V. Biondi, A. Passantino, Clinical practice guidelines: An opinion of the legal implication to veterinary medicine, Animals 9 (8) (2019) 577.

[5] V. Patkar, C. Hurt, R. Steele, S. Love, A. Purushotham, M. Williams, R. Thomson, J. Fox, Evidence-based guidelines and decision support services: a discussion and evaluation in triple assessment of suspected breast cancer, Br. J. Cancer 95 (11) (2006) 1490–1496.

[6] M. Peleg, J. Fox, V. Patkar, D. Glasspool, I. Chronakis, M. South, S. Nassar, J.L. Gaglia, H. Gharib, E. Papini, et al., A computer-interpretable version of the AACE, AME, ETA medical guidelines for clinical practice for the diagnosis and management of thyroid nodules, Endocr. Pr. 20 (4) (2014) 352–359.

[7] A. Ranta, S. Dovey, M. Weatherall, D. O'Dea, J. Gommans, M. Tilyard, Cluster randomized controlled trial of TIA electronic decision support in primary care, Neurology 84 (15) (2015) 1545–1551.

[8] J. Fox, O. Khan, H. Curtis, A. Wright, C. Pal, N. Cockburn, J. Cooper, J.S. Chandan, K. Nirantharakumar, Rapid translation of clinical guidelines into executable knowledge: a case study of COVID-19 and online demonstration, Learn. Heal. Syst. 5 (1) (2021) e10236.

[9] J. Fox, N. Johns, C. Lyons, A. Rahmanzadeh, R. Thomson, P. Wilson, PROforma: a general technology for clinical decision support systems, Comput. Methods Programs Biomed. 54 (1–2) (1997) 59–67.

[10] D.R. Sutton, J. Fox, The syntax and semantics of the PRO forma guideline modeling language, J. Am. Med. Informatics Assoc. 10 (5) (2003) 433–443.

[11] MPS: The domain-specific language creator by JetBrains, 2024, https://www.jetbrains.com/mps/. (Accessed on 07 June 2024).

[12] R. Jepson, D. Brodbelt, C. Vallance, H. Syme, J. Elliott, Evaluation of predictors of the development of azotemia in cats, J. Vet. Intern. Med. 23 (4) (2009) 806–813.

[13] A.H. Sparkes, S. Caney, S. Chalhoub, J. Elliott, N. Finch, I. Gajanayake, C. Langston, H.P. Lefebvre, J. White, J. Quimby, ISFM consensus guidelines on the diagnosis and management of feline chronic kidney disease, J. Feline Med. Surg. 18 (3) (2016) 219–239.

[14] M. Conroy, D.C. Brodbelt, D. O'Neill, Y.-M. Chang, J. Elliott, Chronic kidney disease in cats attending primary care practice in the UK: a VetCompassTM study, Vet. Rec. 184 (17) (2019) 526–526.

[15] IRIS kidney, 2024, http://www.iris-kidney.com/. (Accessed on 07 June 2024).

[16] Interpreting your SDMA results - IDEXX US, 2024, https://www.idexx.com/en/veterinary/reference-laboratories/sdma/interpreting-your-sdma-results/. (Accessed on 07 June 2024).

[17] RenalTech® for kidney disease - Antech diagnostics, 2024, https://www.antechdiagnostics.com/renaltech/. (Accessed on 07 June 2024).

[18] CVS group: Care at our heart | CVS group, 2024, https://www.cvsukltd.co.uk/. (Accessed on 07 June 2024).

[19] CVS knowledge hub – Course catalogue, 2024, https://knowledgehub.cvsukltd.co.uk/. (Accessed on 07 June 2024).

[20] S. Meacham, S. Fraser, G. Ukeje, A. Adedeji, Customised computerised clinical protocol guidelines for medical education: the case of Chronic Kidney Disease (CKD) using domain-specific languages, 2023, https://www.advance-he.ac.uk/programmes-events/conferences/NET2023-Conference#Abstracts.

[21] J. Fox, N. Johns, C. Lyons, A. Rahmanzadeh, R. Thomson, P. Wilson, PROforma: a general technology for clinical decision support systems, Comput. Methods Programs Biomed. (ISSN: 01692607) 54 (1–2) (1997) 59–67, http://dx.doi.org/10.1016/S0169-2607(97)00034-5, URL https://linkinghub.elsevier.com/retrieve/pii/S0169260797000345.

[22] PROformajs, 2024, https://openclinical.gitlab.io/proformajs/. (Accessed on 03 June 2024).

[23] A. Seitinger, A. Rappelsberger, H. Leitich, M. Binder, K.-P. Adlassnig, Executable medical guidelines with Arden syntax—Applications in dermatology and obstetrics, Artif. Intell. Med. 92 (2016) http://dx.doi.org/10.1016/j.artmed.2016.08.003.

[24] Y. Shahar, S. Miksch, P. Johnson, The Asgaard project: a task-specific framework for the application and critiquing of time-oriented clinical guidelines, Artif. Intell. Med. (ISSN: 0933-3657) 14 (1–2) (1998) 29–51, http://dx.doi.org/10.1016/s0933-3657(98)00015-3.

[25] M. Peleg, A. Boxwala, E. Bernstam, S. Tu, R. Greenes, E. Shortliffe, Sharable representation of clinical guidelines in GLIF: Relationship to the Arden Syntax, J. Biomed. Informatics 34 (2001) 170–181, http://dx.doi.org/10.1006/jbin.2001.1016.

[26] openEHR home, 2024, https://openehr.org/. (Accessed on 03 June 2024).

[27] R. Grangel, C. Campos Sancho, B. Martínez-Salvador, M. Marcos, A meta-model for the guideline definition language: in: Proceedings of the 16th International Conference on Software Technologies, SCITEPRESS - Science and Technology Publications, Online Streaming, — Select a Country —, ISBN: 978-989-758-523-4, 2021, pp. 59–68, http://dx.doi.org/10.5220/0010543400590068, URL https://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0010543400590068.

[28] MDENet – Home of the MDE community, 2024, https://mde-network.com/. (Accessed on 03 June 2024).

[29] M. Saxena, S. Song, L. Sha, MediK: Towards safe guideline-based clinical decision support, in: 2023 Formal Methods in Computer-Aided Design (FMCAD), IEEE, 2023, pp. 306–317, URL https://library.oapen.org/bitstream/handle/20.500.12657/85158/Nadel-2023-Proceedings%20of%20the%2023rd%20Conference%20on%20Formal%20Methods%20in%20Comput...-vor.pdf?sequence=1#page=320.

[30] N.K. Singh, Y. Ait-Ameur, D. Méry, Formal ontological analysis for medical protocols, in: Implicit and Explicit Semantics Integration in Proof-Based Developments of Discrete Systems: Communications of NII Shonan Meetings, Springer, 2021, pp. 83–107.

[31] R. Hatko, J. Baumeister, V. Belli, F. Puppe, DiaFlux: A graphical language for computer-interpretable guidelines, in: International Workshop on Knowledge Representation for Health Care, Springer, 2011, pp. 94–107.

[32] B. Kamsu-Foguem, G. Tchuenté-Foguem, C. Foguem, Using conceptual graphs for clinical guidelines representation and knowledge visualization, Inf. Syst. Front. 16 (2014) 571–589.

[33] K. Fox, J. Fox, N. Bexfield, P. Freeman, Computerised decision support in veterinary medicine, exemplified in a canine idiopathic epilepsy care pathway, J. Small Anim. Pr. 62 (10) (2021) 911–917.

[34] P. Pascoe, N. Ducharme, G. Ducharme, J. Lumsden, A computer-derived protocol using recursive partitioning to aid in estimating prognosis of horses with abdominal pain in referral hospitals, Can. J. Vet. Res. 54 (3) (1990) 373.

[35] Feline diabetic diet research trial : Reglucat study - projects - research at the RVC - Royal veterinary college, RVC, 2024, https://www.rvc.ac.uk/research/projects/feline-diabetic-diet-research-trial-reglucat-study?_ga=2.4963571.1557278472.1685468653-1432189476.1685468653. (Accessed on 07 June 2024).

[36] J. Whitlock, J. Dixon, C. Sherlock, R. Tucker, D. Bolt, R. Weller, Technical innovation changes standard radiographic protocols in veterinary medicine: is it necessary to obtain two dorsoproximal–palmarodistal oblique views of the equine foot when using computerised radiography systems? Vet. Rec. 178 (21) (2016) 531–531.

[37] E. Crowther, N. Booth, N. Coombes, S. Baillie, Veterinary clinical skills labs: online collaboration and moving forward, Heal. Soc. Care Educ. 2 (1) (2013) 39–43.

[38] P. Taylor, R. Meyer, Veterinary clinical research or experiments on pets, Vet. Anaesth. Analg. 50 (5) (2023) 383–385.

[39] O.J.J. Heino, The Clinical Decision-Making Process and the Potential Impact of the GekkoVet Clinical Decision Support System on Diagnostics in Veterinary Medicine, Eesti Maaülikool, 2023.

[40] Practice profile: West midlands veterinary referrals | Vet times, 2024, https://www.vettimes.co.uk/article/practice-profile-west-midlands-veterinary-referrals/. (Accessed on 07 June 2024).

[41] A. Seitinger, A. Rappelsberger, H. Leitich, M. Binder, K.-P. Adlassnig, Executable medical guidelines with Arden Syntax—Applications in dermatology and obstetrics, Artif. Intell. Med. 92 (2018) 71–81.

[42] N. Iglesias, J.M. Juarez, M. Campos, Comprehensive analysis of rule formalisms to represent clinical guidelines: Selection criteria and case study on antibiotic clinical guidelines, Artif. Intell. Med. 103 (2020) 101741.

[43] D. O'Sullivan, P. Fraccaro, E. Carson, P. Weller, Decision time for clinical decision support systems, Clin. Med. 14 (4) (2014) 338–341.
[44] JetBrains, MPS: The domain-specific language creator by JetBrains, 2024, https://www.jetbrains.com/mps/. (Accessed on 03 June 2024).
[45] M. Völter, DSL engineering: Designing, implementing and using domain-specific languages, 2013, URL https://voelter.de/dslbook/, Online book. (Accessed 17 November 2024).
[46] V. Pech, JetBrains MPS: Why modern language workbenches matter, in: Domain-Specific Languages in Practice: With JetBrains MPS, Springer, 2021, pp. 1–22.
[47] IRIS kidney - guidelines - IRIS staging of CKD, 2024, http://iris-kidney.com/guidelines/staging.html. (Accessed on 07 June 2024).
[48] O. Boiarskyi, S. Popereshnyak, Automated system and domain-specific language for medical data collection and processing, in: Lecture Notes in Computational Intelligence and Decision Making: 2021 International Scientific Conference "Intellectual Systems of Decision-Making and Problems of Computational Intelligence", Proceedings, Springer, 2022, pp. 377–396.
[49] M. Voelter, J. Siegmund, T. Berger, B. Kolb, Towards user-friendly projectional editors, in: International Conference on Software Language Engineering, Springer, 2014, pp. 41–61.
[50] M. Qadir, Role of Automation in Computer-Based Systems, Citeseer.
[51] CKD-protocol4cats/ at main · stsasakou/CKD-protocol4cats, 2024, https://github.com/stsasakou/CKD-protocol4cats/tree/main. (Accessed on 03 September 2024).
[52] H. Bünder, H. Kuchen, Towards multi-editor support for domain-specific languages utilizing the language server protocol, in: Model-Driven Engineering and Software Development: 7th International Conference, MODELSWARD 2019, Prague, Czech Republic, February 20–22, 2019, Revised Selected Papers 7, Springer, 2020, pp. 225–245.
[53] M. Simi, F. Campagne, Composable languages for bioinformatics: the NYoSh experiment, PeerJ 2 (2014) e241.
[54] S. Chodarev, M. Sulír, J. Porubän, M. Kopčáková, Experimental comparison of editor types for domain-specific languages, Appl. Sci. 12 (19) (2022) 9893.
[55] M. Challenger, G. Kardas, B. Tekinerdogan, A systematic approach to evaluating domain-specific modeling language environments for multi-agent systems, Softw. Qual. J. 24 (2016) 755–795.
[56] P. Runeson, M. Höst, Guidelines for conducting and reporting case study research in software engineering, Empir. Softw. Eng. 14 (2009) 131–164.
[57] S. Meacham, V. Pech, D. Nauck, Classification algorithms framework (CAF) to enable intelligent systems using JetBrains MPS domain-specific languages environment, IEEE Access 8 (2020) 14832–14840.
[58] S. Meacham, V. Pech, D. Nauck, AdaptiveVLE: An integrated framework for personalized online education using MPS JetBrains domain-specific modeling environment, IEEE Access 8 (2020) 184621–184632.
[59] S. Meacham, V. Pech, D. Nauck, AdaptiveSystems: An integrated framework for adaptive systems design and development using MPS JetBrains domain-specific modeling environment, IEEE Access 9 (2021) 127973–127984.
[60] S. Krug, et al., Don't make me think, Revisited, in: A Common Sense Approach to Web and Mobile Usability, 2014.
[61] B.M. McSkimming, S. Mackay, A. Decker, Investigating the usage of Likert-style items within computer science education research instruments, in: 2021 IEEE Frontiers in Education Conference, FIE, IEEE, 2021, pp. 1–8.
[62] J. Saldaña, M. Omasta, Qualitative Research: Analyzing Life, Sage Publications, 2016.
[63] Nvivo leading qualitative data analysis software (QDAS) by lumivero, 2024, https://lumivero.com/products/nvivo/. (Accessed on 14 November 2024).
[64] V.C. Pham, A. Radermacher, S. Gerard, S. Li, Bidirectional mapping between architecture model and code for synchronization, in: 2017 IEEE International Conference on Software Architecture, ICSA, IEEE, 2017, pp. 239–242.
[65] D. Ratiu, B. Schaetz, M. Voelter, B. Kolb, Language engineering as an enabler for incrementally defined formal analyses, in: 2012 First International Workshop on Formal Methods in Software Engineering: Rigorous and Agile Approaches, FormSERA, IEEE, 2012, pp. 9–15.
[66] M. Voelter, B. Kolb, T. Szabó, D. Ratiu, A. van Deursen, Lessons learned from developing mbeddr: a case study in language engineering with MPS, Softw. Syst. Model. 18 (2019) 585–630.
[67] J. Rubin, D. Chisnell, Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests, John Wiley & Sons, 2008.