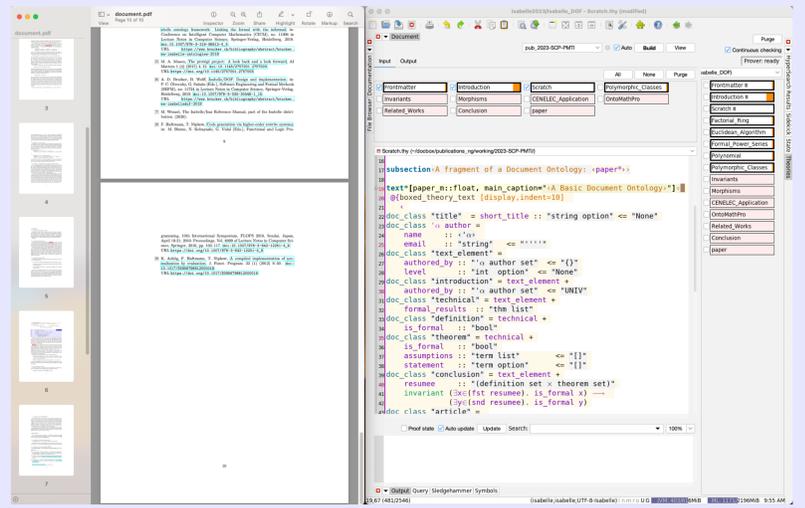




DOF



Document Ontology Framework (DOF), est un environnement :

- pour spécifier des ontologies en ODL, le langage Ontology Document Language
- pour annoter des modèles, λ -termes, preuves, éléments de texte, du code
- pour faire respecter la cohérence entre des théories et leurs documents
- pour soutenir la cohérence entre le formel et l'informel.
- Isabelle/DOF est profondément intégré dans l'écosystème de l'IDE d'Isabelle
- ... permettant un aperçu incrémental.

Isabelle/DOF, ce sont des documents cohérents par validation

Pour les documents qui contiennent à la fois un contenu formel et informel, le lien entre les deux est notoirement difficile à maintenir, surtout si le contenu évolue au cours du développement. Une fois validé par rapport à une ontologie, les définitions, preuves, code annoté et les éléments de texte définis en Isabelle/HOL sont automatiquement convertis vers LaTeX ou HTML. La validation se fait "à la volée" dans l'IDE, et la génération du PDF peut être incrémentale.

DOF est un outil pour écrire des articles scientifiques ...

Relativement similaire à Javadoc, Doxygen ou ocamlDoc, Isabelle/DOF permet l'intégration d'une sorte de macros sémantiques (appelées *antiquotations*) dans des éléments de texte, des λ -termes, des preuves ou du code. L'élément de texte :

```
text⟨According to the reflexivity axiom @⟨thm refl⟩, we obtain in  $\Gamma$ 
for @⟨term (fac 5)⟩ the result @⟨value (fac 5)⟩.⟩
```

est converti après analyse via LaTeX en PDF, soit :

```
According to the reflexivity axiom  $x = x$ , we obtain in  $\Gamma$  for fac 5 the result 120.
```

DOF est un outil pour des rapports techniques ...

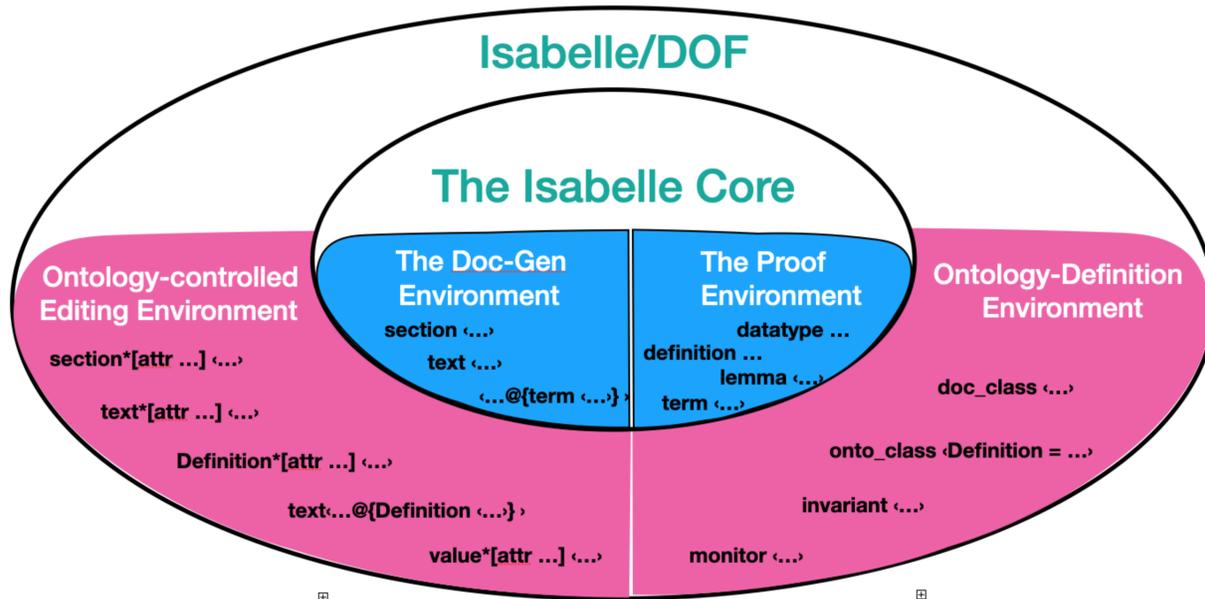
Isabelle/DOF généralise le mécanisme des antiquotations en permettant de les définir via une ontologie. On peut par exemple, à partir d'un standard d'ingénierie logicielle, définir une *ontologie de domaine* et des concepts tels que "Définition informelle", "exigences", "éléments de modèle" et "argument de garantie", etc.

```
text*[safe::Definition, name=safety]
  <Our system is safe if the following holds ...>
```

est référencé après-coup de la manière suivante :

```
text*[...]<As stated in @⟨Definition <safe>⟩, . . . >
```

L'écosystème Isabelle/DOF



DOF et ODL : un langage pour le lien entre le formel et l'informel

Documentation :

- annotations avec validation d'instance
- référencement avec validation de type
- objets d'instance mutables
- évaluation pour filtrage
- monitors pour des validations structurelles
- validation "à la volée" dans l'IDE
- navigation via l'IDE dans le document
- navigation via l'IDE dans l'ontologie

Spécification en ODL :

- héritage des classes, sous-typage
- typage avec polymorphisme ordonné par des "sorts"
- contrats et invariants
- une logique d'ordre supérieure
- un langage fonctionnel de requettes
- types algébriques
- prédicats inductifs

ODL Ontologies et Preuves

- Isabelle/DOF génère plus que des meta-données typées :
- ... le système génère une *théorie* des meta-données en HOL
- Conséquence : preuves sur des morphismes des meta-données
- Conséquence : preuves sur la préservation des invariants
- Conséquence : preuves sur la génération du code
- Les objets d'instances sont des termes dans le λ -calcul, et donc peuvent être intégrés dans tous les λ -termes d'Isabelle
- Conséquence : les objets d'instances peuvent être introduits dans des objets des preuves (théorie "Metalogic" de Nipkow et Roßkopf (AFP)).

Collaborations

Collaborations : • U. de York • IRIT/enseiht Toulouse • U. Paris-Saclay • U. Exeter • U. Nancy • IRT SystemX

Bibliographie

- A. D. Brucker, I. Ait-Sadoun, N. Méric, and B. Wolff, *Using deep ontologies in formal software engineering*, ABZ 2023, LNCS, Springer, 2023, pp. 15–32.
- A. D. Brucker and B. Wolff, *Isabelle/DOF : Design and implementation*, SEFM 2019, LNCS, Springer, 2019, pp. 275–292.