Modélisation déclarative et structurée de problème en vue de la résolution

Pierre-Alain Yvars¹, Laurent Zimmer²

¹ ISAE-Supméca, Quartz, EA 7393, France
pierre-alain.yvars@isae-supmeca.fr

² Dassault-Aviation, Direction de la prospective, France
laurent.zimmer@dassault-aviation.com

Mots-clés : *Modélisation et résolution de problème, DEPS, programmation mathématique linéaire et non linéaire.*

1 Introduction

En matière de modélisation de problème deux politiques sont envisageables:

- Utiliser une librairie de classes prédéfinies dans un langage hôte autorisant d'une part un branchement vers différents solveurs et d'autre part pour certains une structuration du problème en blocs. C'est le cas de Jump avec Julia [1], Pyomo avec python [2] ou bien encore SMS++ avec C++ [3]. L'utilisateur doit ainsi bâtir un programme dans le langage hôte pour expliciter son problème et le résoudre.
- Utiliser un langage de modélisation à part entière, indépendant de tout langage de programmation. Citons les langages de programmation mathématique comme OPL[4] pour la programmation linéaire, AMPL[5] et GAMS [6] pour l'optimisation linéaire ou non linéaire. Toutes ces représentations permettent une modélisation d'un problème à résoudre à l'aide de variables, matrices, équations et/ou inéquations à satisfaire. Ils permettent également la formalisation mathématique d'une fonction objectif à optimiser. Ainsi, les problèmes d'optimisation qu'ils soient linéaires ou non linéaires, réels ou en nombres entiers sont exprimés sur la base d'une vision purement mathématique: un ensemble de variables, un ensemble d'équations et un objectif.

Du point de vue qui nous intéresse, celui de l'ingénieur cherchant à représenter un problème lié à un système technique donné, cette seconde politique souffre d'un manque de capacité de structuration et d'abstraction. En effet celle-ci ne permet de représenter explicitement les structures métier du problème. Le travail présenté vise à combler ce manque c'est-à-dire passer de la représentation métier à la formulation mathématique du problème. Nous présentons quelques extensions récemment apportées à un langage de modélisation structuré de problème de conception de système ainsi qu'à son environnement de modélisation-compilation-résolution en vue d'une généralisation à d'autres problèmes de recherche d'admissibilité ou d'optimalité.

2 Extension du langage DEPS

DEPS [7] est un langage textuel déclaratif à base de modèles et de propriétés. Il autorise une représentation structurée de problèmes de conception de systèmes. Les modèles peuvent-être combinés par héritage, composition et agrégation. Tout modèle DEPS encapsule un ensemble de constantes, de variables, d'éléments (instances d'autres modèles) et de propriétés (équations, inéquations linéaires ou non, propriétés par morceaux, expressions conditionnelles et catalogues) à satisfaire par n'importe quelle instance de ce modèle. Constantes et variables sont typées par des quantités soit prédéfinies soit définies par l'utilisateur (grandeurs, unités, dimension, domaines de valeurs). Les variables peuvent être définies dans des intervalles continus, entiers ou bien dans des domaines énumérés de valeurs entières

ou décimales. Ce langage a été utilisé pour la modélisation de problèmes de dimensionnement, de configuration, de génération d'architecture et de déploiement de fonctions tant pour des systèmes technologiques (mécanique, énergétique, ...) que pour des systèmes à dominante logicielle (systèmes embarqués et systèmes cyber-physiques). DEPS a été récemment étendu pour permettre la prise en compte de collections d'instances de modèles d'éléments, l'utilisation d'opérateurs et la pose de propriétés sur ces collections. L'idée générale est de pouvoir manipuler des collections d'objets métiers plutôt que des tableaux de variables.

3 Extension de l'IDE DEPS Studio

L'environnement intégré de modélisation et de résolution DEPS Studio [8] associé au langage DEPS comprend un éditeur de modèle, un gestionnaire de projet, un compilateur et un solveur. A partir d'un problème modélisé en DEPS, le compilateur de DEPS Studio produit une instance structurée du modèle du problème. L'IDE intègre un solveur de programmation par contraintes en domaine mixte dédié. Trois modes d'exécution sont possibles : Réduction des domaines des variables, recherche d'une ou de plusieurs solutions admissibles, recherche d'une ou de plusieurs solution optimales. Le compilateur a été étendu de manière à pouvoir aussi générer automatiquement un modèle du problème à destination d'un formalisme de programmation mathématique donné. Une première réalisation a été faite avec AMPL. Un modèle écrit en DEPS une fois compilé peut produire un modèle AMPL et permettre ainsi la délégation de sa résolution à un des solveurs d'AMPL en fonction de sa nature (PL, PLNE, MINLP, ...). Le modèle AMPL peut-être produit soit directement par le compilateur soit après une étape de réduction de domaines réalisée par le solveur intégré à l'IDE.

4 Conclusion

Ce travail représente une étape pour modéliser un problème structuré d'optimisation mathématique à l'aide du langage DEPS puis de déléguer sa résolution à des solveurs existants. L'ensemble a été testé sur plusieurs problèmes d'admissibilité et d'optimisation.

Références

- [1] Miles Lubin, Oscar Dowson, Joaquim Dias Garcia, Joey Huchette, Benoît Legat and Juan Pablo Vielma. JuMP 1.0: Recent improvements to a modeling language for mathematical optimization, *Mathematical Programming Computation*, 2023.
- [2] M.L. Bynum, G.A. Hackebeil, W.E. Hart, C.D. Laird, B. Nicholson, J.D. Sirola, J.P. Watson and D.L. Woodruff. *Pyomo – Optimization Modeling in Python*. Third Edition. Springer, 2021.
- [3] <https://gitlab.com/smspp/smspp-project>
- [4] <https://www.ibm.com/docs/en/icos/12.8.0.0?topic=manual-opl-modeling-language>
- [5] Fourer R., Gay D.M., & Kerdighan D.W. *AMPL A language for mathematical programming*. Duxbury & Thomson. 2023. <https://ampl.github.io/ampl-book.pdf>
- [6] Rosenthal R.E. *GAMS a users's guide*. GAMS Development Corporation. Washington, 2007.
- [7] Pierre-Alain Yvars and Laurent Zimmer. DEPS : A model- and property- based language for system synthesis problem, *Software and System Modeling*, 2023. <https://doi.org/10.1007/s10270-023-01129-x>
- [8] Pierre-Alain Yvars and Laurent Zimmer, DEPS Studio : Un environnement intégré de modélisation et de résolution pour la synthèse de système. *20èmes journées Approches Formelles dans l'Assistance au Développement de Logiciels – AFADL*, 2021.