

Teoría de Categorías Aplicada a Variabilidad

Daniel-Jesus Munoz, Monica Pinto y Lidia Fuentes

ITIS Software, CAOSD, Universidad de Málaga, Andalucía Tech, España
{danimg,pinto,lff}@lcc.uma.es,
<http://caosd.lcc.uma.es>

Resumen La *Teoría de Categorías* es un álgebra abstracta que capta los componentes comunes de estructuras aparentemente diferentes. Sus principios organizadores pretenden remodelar y reformular problemas, facilitando su resolución y abriendo puertas a nuevas vías de investigación. En este trabajo analizamos su aplicabilidad a *Modelos de Variabilidad* con los objetivos de estandarizar su expresividad y expandir el conjunto tradicional de herramientas de razonamiento y optimización.

Keywords: teoría de categorías · modelo de variabilidad · característica · razonador · atributo de calidad

1. Introducción

Adoptar nuevas técnicas y algoritmos a la par que unificar los modelos existentes es uno de los retos actuales en variabilidad [9]. Aun considerando que tanto la comunidad de código abierto como la industria han desarrollado eco-sistemas independientes [8], existen tres esfuerzos considerados proto-estándares:

1. Feature Modelling [4]: Modelado, análisis y explotación del dominio.
2. Decision Modelling [8]: Derivación de productos con restricciones.
3. Quality Modelling [1]: Optimización con requisitos de calidad.

Sus *Modelos de Variabilidad* contienen cierta ortogonalidad [7], desde su representación gráfica como árboles de decisión con restricciones cruzadas, a sus conjuntos de variables y operaciones lógicas y aritméticas. No obstante, la tendencia es a extenderlos con características cruzadas y/o de otras disciplinas basadas en modelos. Dada la extensión, no pretendemos detallar todas las soluciones, pero uno de los más representativos sería el ecosistema Clafer [2], donde se integra un lenguaje orientado a objetos con soporte lógico y aritmético, razonadores, optimizadores multiobjetivo, etc. En el caso del framework FAMA [6], resaltamos la incorporación de atributos de características para la generación de modelos de predicción. Otra solución es la fusión de los modelos de decisión y de calidad directamente en los modelos de características [12]. Por otra parte, el eco-sistema HADAS [14] es un híbrido entre Clafer y una base de datos relacional para resolver problemas de requisitos no funcionales. Por último, encontramos la resolución de modelos de calidad con un enfoque de desarrollo dirigido por modelos [11]. En definitiva, la *reinención de la rueda* [13].

Para convertir en ventajas una situación similar, surgió en los '70s la *Teoría de Categorías* (TC). TC es una teoría matemática general sobre estructuras algebraicas que permite captar y relacionar los aspectos comunes de diferentes estructuras, abstrayéndose de sus particularidades [3]. De manera natural, se usan los conceptos categóricos en la transferencia e integración de conocimientos entre disciplinas, incluyendo lenguajes, modelos, razonadores, etc. Resumidamente, una *Categoría* es cualquier colección de *Objetos* que representan espacios que pueden relacionarse entre sí mediante *Flechas* (es decir, *Morfismos*), cumpliendo composición asociativa e identidad. *Set* es un ejemplo donde los objetos son conjuntos y las flechas son funciones entre conjuntos. De hecho, en [5] se propone un marco de diseño computacional basado en objetos y flechas, el cual se expande como transformaciones categóricas de modelos en [16].

En este trabajo **proponemos** cohesionar categóricamente el modelado de variabilidad, donde cada característica sea un objeto categórico, y sus interrelaciones y operaciones sean flechas, ya sea internamente entre los elementos de un objeto (e.g., relación padre-hijo) o cruzando objetos (e.g., atributo de calidad de un producto completo). Las **contribuciones** potenciales son (1) los investigadores y los profesionales pueden extender su casuística sin perder el trabajo existente, (2) utilizar, experimentar e integrar las herramientas existentes, (3) adaptar casuísticas concretas hacia razonadores más eficientes, y (4) desarrollar un enfoque de modelado de variabilidad estándar.

2. Visión General

En la Tabla 1 se muestra la sintaxis en variabilidad y TC, lo cual nos permite comparar sus diferentes capacidades, así como visualizar equivalencias en el caso de que existan. Podemos visualizar que TC es más extenso y flexible, permitiendo la composición de sub-esquemas y un dominio variable más extenso considerando números reales, cadenas, etc. Basándonos en esto detallamos en la Figura 1 un modelo de variabilidad categorizado a modo de ejemplo, el cual es un multigrafo con forma de árbol para aumentar la similitud con los árboles de variabilidad. El ejemplo consta de 5 objetos categóricos, donde 3 son submodelos (sub-árboles) y 2 son dominios de datos a modo de *nombre:valor* (i.e., variables Booleanas o enteras). Cada nodo es un elemento de un objeto, y sus relaciones, ya sean entre elementos u objetos, se definen con una flecha. Los tipos de flecha cubren dominio, parentesco y restricciones cruzadas, pero su potencial permitiría relacionar directamente productos completos, atributos, calidad, etc. Ésta categoría es compatible con el entorno de desarrollo integrado *Categorical Query Language* (CQL) IDE [15], un software funcional canónico que genera multígrafos como el de la Figura 1, y que contiene razonadores de diferentes disciplinas que trabajan de manera **concurrente** y **eficiente**: un demostrador automático de teoremas basado en el algoritmo de finalización de Knuth-Bendix para lógica de orden superior y ecuaciones aritméticas, y hashing, árboles binarios de búsqueda y algoritmos de persecución de objetivos para relaciones, restricciones y requisitos. Otra opción es la bifurcación de CQL denominada Statebox, la cual

Tabla 1. Comparativa sintáctica de las disciplinas *Variabilidad* y *Teoría de Categorías*

→Disciplina	Variabilidad	Teoría de Categorías
↓ <i>Funcionalidad</i>		
Esquema	Diagrama/Árbol	Categoría/Grafo
<i>Componentes</i>	Nodos y Productos	Sub-categorías, Objetos e Instancias
<i>Dominio</i>	{ \mathbb{B}, \mathbb{Z} }	{ $\mathbb{B}, \mathbb{N}, \mathbb{Z}, \mathbb{R}, \dots$ }
<i>Restricciones</i>	Relaciones y ecuaciones	Flechas y Funtores
<i>Selección</i>	Requisito y exclusión	$(-)\Delta$
<i>Conectivas</i>	{ $\&\&, , [x.y], \Rightarrow, \Leftrightarrow$ }	{ $\sum_{[Funcioner]} \prod_{[Funcioner]}, X$ }
<i>(Des)Igualdades</i>	{ $=, \neq, >, \geq, <, \leq$ }	{ $=, \neq, >, \geq, <, \leq$ }
<i>Aritmética</i>	{ $+, -, *, \div, \%, \dots$ }	{ $+, -, *, \div, \%, \dots$ }

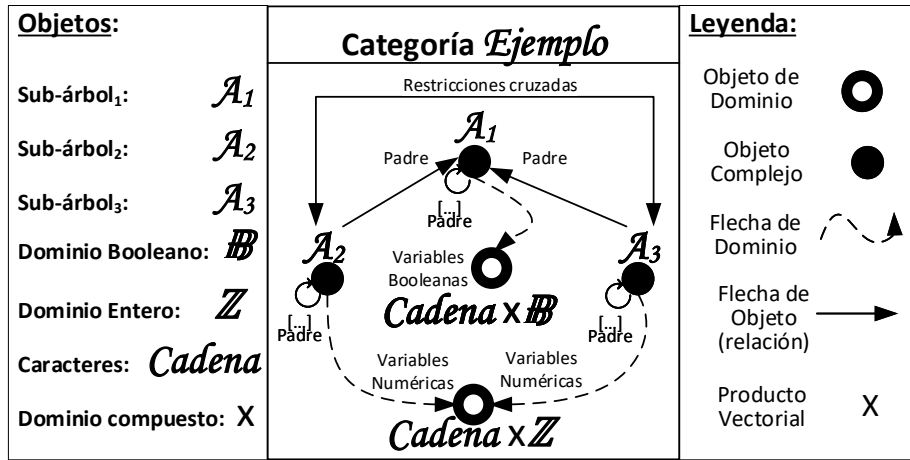


Figura 1. Ejemplo de *Modelo de Variabilidad* categórico con 3 árboles y 2 dominios

presenta un lenguaje visual para el modelado composicional de procesos expandiendo CQL mediante redes Petri, Teoría de Juegos y criptografía [10]. Ambos entornos tienen planificado integrar nuevos solvers que se seguirán ejecutando a medida eficientemente y de manera transparente al desarrollador.

3. Conclusiones

En este artículo de prospección hemos evaluado la necesidad y potenciales ventajas de un framework categórico para cohesionar el modelado de variabilidad con requisitos y optimización de productos. Así mismo, hemos presentado una visión general de la transformación de nodos/relaciones/productos en variabilidad a objetos/flechas/instancias en TC y las herramientas para evaluarla.

Agradecimientos

Este trabajo está financiado por el programa de investigación e innovación H2020 de la Unión Europea bajo el acuerdo de subvención DAEMON 101017109,

por los proyectos también co-financiados por fondos FEDER LEIA UMA18-FEDERJA-15, MEDEA RTI2018-099213-B-I00 y Rhea P18-FR-1081, y la ayuda PRE2019-087496 del Ministerio de Ciencia e Innovación.

Referencias

1. Al-Qutaisi, R.E.: Quality models in software engineering literature: an analytical and comparative study. *Journal of American Science* **6**(3), 166–175 (2010)
2. Bak, K., Diskin, Z., Antkiewicz, M., Czarnecki, K., Wasowski, A.: Clafer: unifying class and feature modeling. *Software & Systems Modeling* **15**(3), 811–845 (2016)
3. Barr, M., Wells, C.: *Category theory for computing science*. Prentice Hall (1990)
4. Batory, D.: Feature models, grammars, and propositional formulas. In: Obbink, H., Pohl, K. (eds.) *Software Product Lines*. pp. 7–20. Springer, Berlin (2005)
5. Batory, D., Azanza, M., Saraiva, J.: The objects and arrows of computational design. In: *International Conference on Model Driven Engineering Languages and Systems*. pp. 1–20. Springer (2008)
6. Benavides, D., Trinidad, P., Ruiz-Cortés, A., Segura, S.: FaMa, pp. 163–171. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
7. Budiardjo, E.K., Zamzami, E.M., et al.: Feature modeling and variability modeling syntactic notation comparison and mapping. *Computer and Communications* ('14)
8. Czarnecki, K., Grünbacher, P., Rabiser, R., Schmid, K., Wkasowski, A.: Cool features and tough decisions: A comparison of variability modeling approaches. In: *Proceedings of the Sixth International Workshop on Variability Modeling of Software-Intensive Systems*. p. 173–182. ACM, New York, NY, USA (2012)
9. Feichtinger, K., Rabiser, R.: Variability model transformations: Towards unifying variability modeling. In: *2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. pp. 179–182 (2020)
10. Genovese, F., Spivak, D.I.: A categorical semantics for guarded petri nets. In: Gadducci, F., Kehrer, T. (eds.) *Graph Transformation*. pp. 57–74. Springer International Publishing, Cham (2020)
11. González-Huerta, J., Insfran, E., Abrahão, S., McGregor, J.D.: Non-functional requirements in model-driven software product line engineering. In: *Proceedings of the Fourth International Workshop on Nonfunctional System Properties in Domain Specific Modeling Languages. NFPinDSML '12, NY, USA* (2012)
12. Horcas, J.M., Pinto, M., Fuentes, L.: An automatic process for weaving functional quality attributes using a software product line approach. *Journal of Systems and Software* **112**, 78 – 95 (2016)
13. Llitas, A.B., Torres, D., Collazos, C.A., Fernandez, A.: Development, reuse, and repurposing of software artifacts in digital citizen science. are we reinventing the wheel? In: *CEUR Workshop Proceedings* (2020)
14. Munoz, D.J., Pinto, M., Fuentes, L.: Hadas: analysing quality attributes of software configurations. In: *Proceedings of the 23rd International Systems and Software Product Line Conference-Volume B*. pp. 13–16 (2019)
15. SCHULTZ, P., WISNESKY, R.: Algebraic data integration. *Journal of Functional Programming* **27**, e24 (2017)
16. Taentzer, G., Salay, R., Strüber, D., Checkik, M.: Transformations of software product lines: A generalizing framework based on category theory. In: *20th Int. Conference on Model Driven Engineering Languages and Systems (MODELS)* (2017)