Chapter
1

Module and Modularity in Software Engineering

Isaque de Oliveira Dornelas Barbosa – Departamento Acadêmico de Computação – Universidade Tecnológica Federal do Parana (UTFPR)

Abstract

Modularity plays a crucial role in software engineering, significantly impacting the maintainability, extensibility, and adaptability of complex systems. Despite its widespread adoption, the literature lacks consistent definitions and comprehensive studies on modularity, particularly regarding its non-functional aspects and the interplay between different levels of granularity. This research aims to address these gaps by providing a software development-focused definition of modularity, incorporating functional and non-functional attributes such as flexibility, evolvability, testability, and maintainability. The study employs a systematic literature review to identify existing definitions and metrics, complemented by interviews with industry professionals and the development of a modularity assessment tool. The research also explores the impact of organizational culture and interface design on modularization. The expected outcomes include a robust theoretical and practical framework for implementing modularity across various system levels, demonstrating the benefits of modularity in reducing development and maintenance costs, enhancing flexibility, and improving adaptability to new technologies and requirements. This work seeks to promote the standardization of modularity-based software development processes and optimize the value of modular systems in dynamic environments.

Resumo

Modularidade desempenha um papel crucial na engenharia de software, impactando significativamente a manutenibilidade, extensibilidade e adaptabilidade de sistemas complexos. Apesar de sua adoção generalizada, a literatura carece de definições consistentes e estudos abrangentes sobre modularidade, especialmente em relação aos seus aspectos não funcionais e as interações entre diferentes níveis de granularidade. Esta pesquisa visa preencher essas lacunas fornecendo uma definição de modularidade centrada no desenvolvimento de software, que incorpora atributos funcionais e não funcionais, como flexibilidade, evoluibilidade, testabilidade e manutenibilidade. A pesquisa utiliza uma *revisão sistemática da literatura para identificar definições e métricas existentes, complementada por entrevistas com profissionais da indústria e o desenvolvimento de uma ferramenta de avaliação de modularidade. A pesquisa também explora o impacto da cultura organizacional e do design de interface na modularização. Os resultados esperados incluem um robusto quadro teórico e prático para implementar modularidade em diferentes níveis de sistemas, demonstrando benefícios como redução de custos de desenvolvimento e manutenção, maior flexibilidade e melhor adaptabilidade a novas tecnologias e requisitos. Este trabalho busca promover a padronização de processos de desenvolvimento de software baseados em modularidade e otimizar o valor de sistemas modulares em ambientes dinâmicos.*

Keywords: Module, Modularity, Software Engineering, Product Design, Interface.

1.1. Related Work

Software, like any complex system, has a life cycle that we endeavor to extend as much as possible. Maintainability, extensibility, evolvability and ease of understanding of the code are crucial factors in achieving this goal.

The growing interest in modular systems has been significantly driven by the rise of cloud computing, which has facilitated the separation of systems and propelled the adoption of microservices. However, modular systems extend beyond mere physical division, encompassing both microservices and monolithic architectures, with the latter offering advantages in deployment, maintenance, and performance in certain scenarios [Bonvoisin et al. 2016, Efatmaneshnik et al. 2020, Moon and Howison 2024, Prime Video Tech 2024]. Despite this widespread adoption, the literature lacks in-depth studies on the impact of modularity on software systems and the best practices for module formation [Bonvoisin et al. 2016, Morgan et al. 2021]. [Moon and Howison 2024] highlights that many purported benefits of modularity are often assertions without sufficient evidence or argumentation, which complicates decision-making related to modularization.

A module is defined not only by its functionality but also by its non-functional aspects, such as flexibility, evolvability, maintainability, testability, and maintainability. A module can be understood as a cohesive set of components, subsystems, or both, delimited by a clear boundary [Efatmaneshnik et al. 2020]. However, the literature often fails to distinguish between different levels of granularity in system modularity, focusing primarily on architectural modularity while neglecting more detailed levels such as domains, classes, and functions [Efatmaneshnik et al. 2020, Morgan et al. 2021]. Addressing this gap is crucial for optimizing system design and understanding the interactions between modules at various granularity levels.

The concept of modularity is predicated on the necessity to effectively navigate the intrinsic complexities associated with software systems. It provides a means to achieve cost savings in maintenance and to facilitate future development through the decomposition of systems into smaller, independent units [Bonvoisin et al. 2016, Morgan et al. 2021]. However, the process of modularization requires careful consideration of various factors, such as the definition of interfaces between modules to ensure interoperability and avoid unwanted dependencies [Sonego and Echeveste 2015, Bonvoisin et al. 2016]. The selection of appropriate metrics, such as cohesion and coupling, is also crucial for effective module creation [Sonego and Echeveste 2015, Bonvoisin et al. 2016, Efatmaneshnik et al. 2020, Morgan et al. 2021, Moon and Howison 2024]. [Bonvoisin et al. 2016] emphasizes the importance of reproducibility in these metrics to ensure consistent evaluation of component similarity and dependency.

Modularity extends beyond structural aspects to include functional and business dimensions, influencing different levels of a system [Efatmaneshnik et al. 2020]. However, existing research tends to focus on architectural modularity, thereby neglecting its importance at more detailed levels [Efatmaneshnik et al. 2020, Morgan et al. 2021]. It is imperative that modularity research extend beyond architectural modularity to encompass its implementation at increasingly detailed levels. The development of methodologies and metrics for the assessment and enhancement of modularity at the level of domains, classes, and functions represents a crucial step in the creation of more resilient, flexible, and adaptable software systems [Xiang et al. 2019, Efatmaneshnik et al. 2020].

Interfaces play a crucial role in modularity, defining how modules interact and communicate [Sullivan et al. 2001, Efatmaneshnik et al. 2020, Morgan et al. 2021]. Well-defined interfaces promote independence and low coupling, allowing modules to be developed, tested, and modified in isolation [Sullivan et al. 2001, Morgan et al. 2021]. However, poorly defined interfaces can lead to increased coupling and compromise system quality [Bonvoisin et al. 2016].

The absence of standardization in interfaces makes it challenging to reuse modules, constraining the flexibility and adaptability of the system to accommodate new requirements [Efatmaneshnik et al. 2020]. Despite emphasizing the significance of interface standardization, [Efatmaneshnik et al. 2020] acknowledge the existence of benefits associated with non-standardized interfaces. In certain instances, particular interfaces may prove more conducive to optimizing non-functional attributes, such as recycling, maintenance, reuse, and adaptability [Sonego and Echeveste 2015, Efatmaneshnik et al. 2020].

[Sonego and Echeveste 2015] highlights that the way products are divided into modules and the degree of modularity directly influence the benefits a company can achieve. Formal methods, such as the Design Structure Matrix (DSM), can significantly reduce time and resource costs in modular design [Sonego and Echeveste 2015, Efatmaneshnik et al. 2020]. However, modularity is not a one-size-fits-all solution and presents challenges, particularly in rapidly changing technological environments, where its benefits may be limited [Efatmaneshnik et al. 2020].

Recent studies in product design modularity focus on creating reusable modules that enhance business functionality, similar to how software design modularity aims to improve organization and development [Baldwin and Clark 2006, Bonvoisin et al. 2016]. The DSM is a pivotal tool in both software and physical product development, providing a visual representation of system dependencies and facilitating strategic decision-making in modular design [Sullivan et al. 2001, Baldwin and Clark 2006, Efatmaneshnik et al. 2020]. It aids in identifying modular structures, detecting excessive coupling, reorganizing systems, simulating changes, and optimizing team communication [Sullivan et al. 2001].

The influence of organizational culture on modularization is another underex-

Study/ Research	Metrics/ Methods	Functional perspective	Non-functional perspective	Product design focus	Software development focus
[Moon and Howison 2024]	DSM, coupling and information hiding	Yes	Yes	No	Yes
[Alho 2023]	None	Yes	Yes	No	No
[Morgan et al. 2021]	Container modeling method (CMM)	Yes	Yes	Yes	No
[Efatmaneshnik et al. 2020]	None	Yes	Yes	No	No
[Xiang et al. 2019]	Feature coupling network	No	Yes	No	Yes
[Bonvoisin et al. 2016]	Coupling, cohesion and others	Yes	Yes	Yes	No
[Sonego and Echeveste 2015]	DSM, modular function deployment (MFD), fuzzy logic based (FLB) and others	Yes	Yes	Yes	No
[Baldwin and Clark 2006]	DSM, hierarchy, economic value and others	Yes	Yes	Yes	No
[Sullivan et al. 2001]	DSM, flexibility and net options value (NOV)	Yes	Yes	No	Yes
This study	*	Yes	Yes	No	Yes

Table 1.1. Comparative table of related works with the present study. The metrics or methods employed are compared, along with whether they adopt a functional perspective, a non-functional perspective, a focus on product design, and a focus on software development. The use of the symbol * indicates the union of metrics and methods from the presented works.

plored area. Organizational rigidity, lack of communication, and resistance to change can hinder the effectiveness of modular approaches [Sullivan et al. 2001, Moon and Howison 2024]. Understanding how organizational culture can be aligned with modular principles is essential for creating resilient and adaptable systems.

Traditionally, modularization research has focused on functional aspects and early decision-making for module creation, often prioritizing functional decomposition over adaptability and evolution [Morgan et al. 2021]. This study aims to shift the focus towards the adaptability and evolvability of systems, which are critical for systems in development. It seeks to provide clearer definitions of modules and modularity, incorporating both functional and non-functional aspects, and to evaluate existing metrics for their applicability at different granularity levels.

In conclusion, while modularity offers numerous benefits, its effective implementation requires a nuanced understanding of its challenges and the interplay between organizational culture, interface design, and granularity levels. It is anticipated that this study will facilitate the development of tools and metrics to optimize modularity across these dimensions, thereby ensuring that systems remain adaptable and evolvable in dynamic environments.

References

[Alho 2023] Alho, R. (2023). A systematic literature review on the modularity of modular neural networks and comparison to monolithic solutions. Master's dissertation, University of Helsinki, Helsinki, Finland. Accessed: 2024-11-12.

[Baldwin and Clark 2006] Baldwin, C. Y. and Clark, K. B. (2006). *Modularity in the Design of Complex Engineering Systems*, page 175–205. Springer Berlin Heidelberg.

[Bonvoisin et al. 2016] Bonvoisin, J., Halstenberg, F., Buchert, T., and Stark, R. (2016). A systematic literature review on modular product design. *Journal of Engineering Design*, 27(7):488–514.

[Efatmaneshnik et al. 2020] Efatmaneshnik, M., Shoval, S., and Qiao, L. (2020). A standard description of the terms module and modularity for systems engineering. *IEEE Transactions on Engineering Management*, 67(2):365–375.

[Moon and Howison 2024] Moon, E. and Howison, J. (2024). A dynamic perspective on software modularity in open source software (oss) development: A configurational approach. *Information and Organization*, 34(1):100499.

[Morgan et al. 2021] Morgan, M., Holzer, T., and Eveleigh, T. (2021). Synergizing model-based systems engineering, modularity, and software container concepts to manage obsolescence. *Systems Engineering*, 24(5):369–380.

[Prime Video Tech 2024] Prime Video Tech (2024). Scaling up the prime video audio/video monitoring service and reducing costs by 90%. Accessed: 2024-03-25.

[Sonego and Echeveste 2015] Sonego, M. and Echeveste, M. S. (2015). Seleção de métodos para modularização no desenvolvimento de produtos: revisão sistemática. *Production*, 26(2):476–487.

[Sullivan et al. 2001] Sullivan, K. J., Griswold, W. G., Cai, Y., and Hallen, B. (2001). The structure and value of modularity in software design. *SIGSOFT Softw. Eng. Notes*, 26(5):99–108.

[Xiang et al. 2019] Xiang, Y., Pan, W., Jiang, H., Zhu, Y., and Li, H. (2019). Measuring software modularity based on software networks. *Entropy*, 21(4):344.