School of Mathematics, Statistics and Computer Science

CONNECT
To create more value

Axiomatic Design and Axiomatic Software Architecture

Dr.-Ing Aydin Homay
Chair of Industrial Communication, Technische Universität Dresden
Member of IEEE Computer Society
Member of IEEE Industrial Electronics
Life Science and Healthcare Solution Architect at EPAM Systems

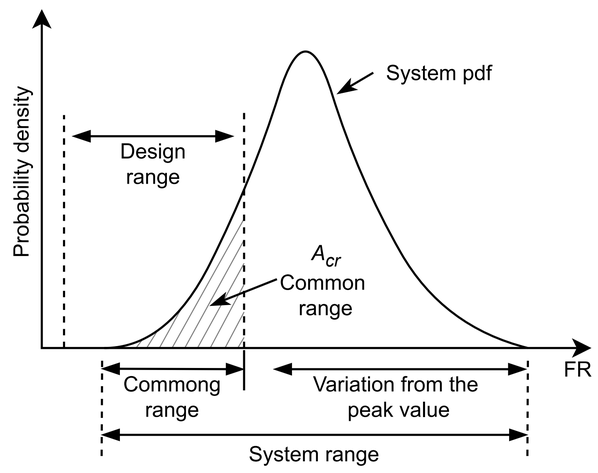
1. Axiomatic Design

Axiomatic Design (AD) is a systematic design based on axioms that helps designers to first identify the problem in the business domain and the requirements in the functional domain using solution-neutral thinking and develop a design solution that effectively solves the identified problems based on axioms and corollaries. AD was developed by Professor Nam P. Suh from MIT to provide a scientific foundation for design. Its core goal is to improve the design process by offering a rational framework that transforms customer needs into effective and efficient design solutions. AD helps designers (i) identify the problem as defined by the customer (in the customer domain), (ii) derive the functional requirements (FRs) in the functional domain using solution-neutral thinking, and (iii) develop a design solution by selecting appropriate design parameters (DPs) in the physical domain in a way that satisfies the FRs while adhering to foundational design principles [2].

2. Axiomatic Software Architecture

Axiomatic software architecture (ASA) based on axiomatic design promotes a holistic view of the system under design and guides the architect through a structured, incremental, and iterative design process. Anti-coupled architecture is at the core of ASA. This approach inherently reduces system complexity, fosters creativity, minimizes reliance on trial-and-error, and enables the identification of optimal design solutions among alternatives.

The ultimate goal of axiomatic software architecture is to shift the start of the design process from the traditional point—after requirements have been formally established—to a much earlier phase. In this early phase, the software architect systematically translates vague, conflicting, and often fuzzy business needs into a minimal set of independent functional requirements. This is achieved by iteratively mapping between "what we want to achieve" and "how we can achieve it," thereby addressing the issue of coupled design within the functional domain, rather than detecting it later in the physical domain, as is commonly done in software practice [1].



در راستای تفاهم‌نامه همکاری با متخصصان و دانشمندان ایرانی غیرمقیم، دانشگاه تهران و معاونت علمی و فناوری ریاست جمهوری، سخنرانی دکتر آیدین همای از دانشگاه فنی درسدن در حوزه علوم کامپیوتر را روزهای شنبه و یکشنبه، ۱۳ و ۱۴ اردیبهشت ۱۴۰۴، ساعت ۱۵ تا ۱۸ در دانشکده ریاضی، آمار و علوم کامپیوتر برگزار می‌کنند. روز اول کارگاه در رابطه با Axiomatic Design و روز دوم کارگاه در رابطه با Axiomatic Software Architecture خواهد بود.

Aydin Homay is a software engineer and researcher in the field of industrial control and automation systems. He earned a Bachelor of Engineering in Software Engineering Technologies (in Tabriz, Iran) and a Master of Science in Artificial Intelligence (in Tehran, Iran). He has recently defended his Ph.D. in Computer Science at the Technical University of Dresden, focusing on flexible industrial automation and control systems. His doctoral research is centered on systematic design for industrial control and automation systems. Aydin has occupied various positions, including junior, senior, and lead software engineer and solution architect at companies such as Baker Hughes, Siemens Healthineers, and Amazon Web Services.

Aydin Homay's career bridges academia and industry in the realm of intelligent industrial systems. He has progressed from developing control software in energy and manufacturing sectors to architecting complex healthcare and cloud solutions. In each role – whether as a junior engineer or a lead architect. Aydin has made significant contributions, on the way that we should design critical systems, how to understand design and complexity in software systems, and how to enhance industrial programming tools, for effective automation and control system design and development.

References

[1] Aydin Homay and Mario De Sousa. Axiomatic Software Architecture. In *22nd IEEE International Conference on Software Architecture (ICSA 2025)*, page 5, Odense, 2025. IEEE.

[2] Nam Pay Suh. *Axiomatic Design Advances and Applications*. Oxford University Press, 2001.