# A Systematic and Efficient Approach to the Design of Modular Hybrid AI Systems

Thomas Schmid[1,2,3]

[1]Martin-Luther-Universität Halle-Wittenberg, Germany
[2]Lancaster University in Leipzig, Germany
[3]Universität Leipzig, Germany

**Abstract**

In recent years, combining machine learning and knowledge engineering has been gaining significant attention among researchers in industry and academia. While many classical hybrid architectures follow a unified or transformative approach, modular architectures built from independent yet interacting modules are becoming more and more popular in practice. State-of-the-art architectures like the IBM Debater system, for example, are based on distinct modules that implement diverse capabilities, such as speech recognition, natural language processing, reasoning, and speech synthesis. For designing future systems of this kind and complexity, we propose to use the so-called AI=MC$^2$ Taxonomy as a systematic and efficient tool. To this end, we introduce a practical step-by-step procedure for specifying modules of hybrid architectures based on specific artificial intelligence (AI) methods and capabilities. We explain this procedure by example and conclude that this approach will be relevant for future AI designs.

**Keywords**

Hybrid Artificial Intelligence, Modular Hybrid Architectures, AI System Design, AI Standardization, AI=MC$^2$ Taxonomy

## 1. Motivation

Modular design is a key principle in modern industrialized development processes. In product design, for example, applying a modularization strategy will yield modular products, or in other words, products that “are made of modules, building blocks“[1]. A module or building block in this sense may be understood as a group of ”functional carriers” like components or parts [2]. The intention therein is to built a product, process or service from several distinct functional carriers in order to create a more comprehensive and complex functionality based on a “construction kit” of well-defined and reusable modules [3]. In architecture and manufacturing, the manifold advantages of modular design – such as decreased development time, increased interoperability and better planning – have been recognized widely, from the Roman Empire [4] to the twentieth century design pioneers of the German Bauhaus arts school [5] all the way to current megaprojects such as Tesla’s giant factories for electric cars [6].

*In A. Martin, K. Hinkelmann, H.-G. Fill, A. Gerber, D. Lenat, R. Stolle, F. van Harmelen (Eds.), Proceedings of the AAAI 2023 Spring Symposium on Challenges Requiring the Combination of Machine Learning and Knowledge Engineering (AAAI-MAKE 2023), Hyatt Regency, San Francisco Airport, California, USA, March 27-29, 2023.*

*Corresponding author.
thomas.schmid@medizin.uni-halle.de (T. Schmid)

In computer science, too, modularization is an influential and widely applied concept, both in hardware and software development. A prominent example is object-oriented programming where organizing all structures and processes in so-called objects represents the central design paradigm [7]. While this may already be viewed as a way of modular design by itself, also specific modules or combinations of interacting objects have been proposed as so-called design patterns [8]. In artificial intelligence (AI), however, rather monolithic approaches that aim at solving even extremely complex tasks, such as generating text [9] or images [10], with a single technique constitute still the dominating paradigm. An exception is the field of hybrid AI, where two or more paradigms of AI with complementing strengths are combined in order to optimize the resulting hybrid AI system [11]. While hybridization within a single system, also termed either unified or transformative [12], has been of particular interest for many researchers [13], today practioners face a growing industry demand for modular hybrid AI approaches. The success of IBM's Debater system, for example, is based on an efficient combination of a variety of sophicated existing AI techniques in modules for distinct subtasks, such as speech recognition, natural language processing, reasoning, and speech synthesis [14].

Modular hybrid AI has been characterized by various criteria. In classical taxonomies, only loosely and tightly coupled hybrid modular architectures are distinguished [12]. More current taxonomies, such as van Bekkum's modular design patterns for hybrid learning and reasoning systems [15], represent more details and complexity and introduce modules similar to design patterns in object-oriented programming. Here, however, we generalize such ideas beyond the relatively narrow field of neuro-symbolic AI and rethink modularization as a comprehensive concept applicable basically with any type of AI techniques. In particular, we expect more large-scale modular hybrid AI systems comparable to the IBM Debater system to be developed in the future and propose a novel strategy to design such systems in a systematic and efficient fashion using the so-called AI=MC$^2$ Taxonomy [16]

## 2. Designing Modular Hybrid AI with the AI=MC$^2$ Taxonomy

### 2.1. Overview of the Taxonomy

Over the last decade, AI-based products and services have become a major technological trend. The development of such AI applications in large numbers and for a growing number of use cases has begun to drive a process of standardizing and modularizing AI solutions in industrial contexts. A prominent exhibit for this the development is the German national roadmap for standardization of AI [17] and its goals: At the end of this process, modularized components and cross-industry integration are intended to be a matter of every-day practice. In order to provide suitable description and design tools for this goal, we have recently established a comprehensive taxonomy of AI methods and capabilities [18, 16, 19], which in a third dimension also includes levels of risk or criticality, respectively (cf. Fig. 1). Termed AI=MC$^2$, this taxonomy is not only a core element of the German national roadmap for standardization of AI [17], but has also been proposed as basis for planned European AI standards.

The AI=MC$^2$ taxonomy reflects and condenses, in particular, available AI methods and thereby implementable cognitive capabilities into a unified framework. A unique strength is its level of granuality: It differentiates not only between top-level capabilities (sense, process, act,

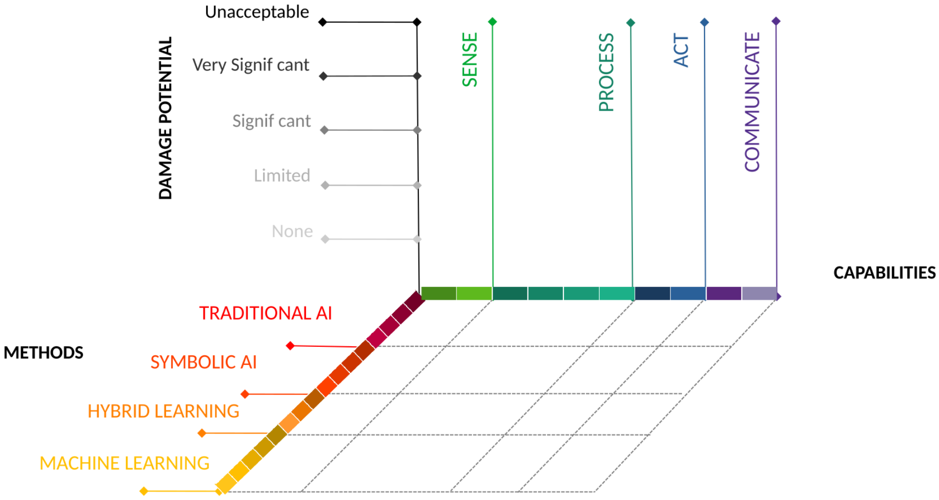


**Figure 1:** Visual representation of the AI=MC$^2$ Taxonomy in a three-dimensional display. Next to AI methods and capabilities, risk or criticality of modules may be represented in a third dimension.

communicate) and top-level methods (traditional AI, symbolic AI, machine learning, hybrid learning), but rather goes into three levels of details on the basis of existing scientific findings and taxonomies; examplarily named specific algorithms make up even for a fourth level. For the capability to process, for example, the subcapability to provide cognitive processing abilities (one among several) is further differentiated into 24 specific and distinct subsubcapabilities. In fact, this taxonomy is today so comprehensive that its most recent version it now covers more than 30 pages in the accompanying book [19]. Another central strength is that due to its systematic and hierarchical nature, the AI=MC$^2$ taxonomy provides an easy and efficient visual approach to understanding and designing modular systems (cf. Fig. 1).

## 2.2. Designing Modular Hybrid AI

Designing by means of modularization may be described as carrying out one or more of the following and potentially complementary tasks [20]: identifying modules, design of modules and design with modules. While designing with predefined modules is a relatively straight forward process, identifying and designing modules typically is not. The underlying challenge in this is to consider a given set of design goals first, and then identify the relevant criteria for clustering components and functions into modules [2]. For both, identifying and designing modules for hybrid AI systems, the AI=MC$^2$ Taxonomy represents an ideal toolkit.

The identification of modules is supported by the taxonomy's dimension of AI capabilities, which allows to differentiate overall goals in functionality and specify subfunctionalities to be implemented in individual modules. The actual design of such modules on the other hand is supported by the dimension of AI methods, which allows an intuitive review and selection from a large variety of techniques available for implementation. In the next section, we will outline how to use the the AI=MC$^2$ Taxonomy efficiently along the entire process from identifying modules to designing modules and finally to designing with specified modules.

## 3. A Step-by-Step Procedure for Designing Modular Hybrid AI

In order to achieve a structured approach, we propose a three-steps procedure for the development process. Initially, a detailed *capabilities analysis* is performed. In a second step, a *methods analysis* is to be carried out. Finally, the interconnections between and/or the ordering of the thereby identified modules is to be determined. This is referred to as the *module pathway*. Going through these three steps provides a straight-forward and efficient route from analysis to implementation. In the following, we illustrate this procedure along an example, in which we aim to design the modules of an audio chatbot comparable to Apple's Siri or Amazon's Alexa. We start under the assumption that no suitable modules have been designed so far.

### 3.1. Step 1: Capabilities Analysis

Similar to requirements analyses in software development, a thorough and structured description of the overall intended functionality is required in order to lay the foundation for an efficient specification of the building blocks of the system. In particular, this implies to translate general requirements for the system into the more specific capabilities defined by the AI=MC$^2$ taxonomy. The outcome of this step is the identification of the required modules.

For our example of designing an audio chatbot, we would in this step start with the rather general observation that is should be able to sense, process and communicate in order to provide the intended functionality. From these top-level capabilities, second-level capabilities are identified. For sense, for example, the second level would make a distinction between internal and external sensing, where internal comprises self-awareness and balance and external is further divided into the capabilities to see, hear, small, taste, and touch (Fig. 2). The most specific capability – here: hearing – then identifies our first module.

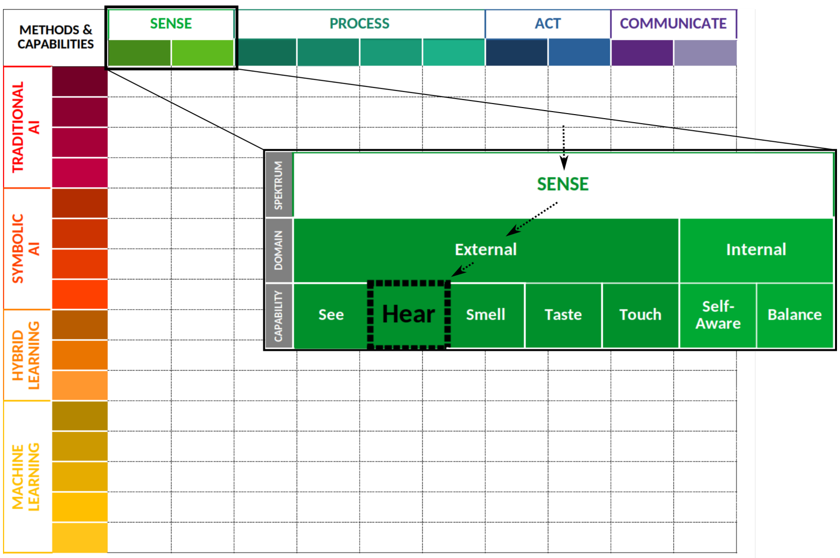


**Figure 2:** Capabilities analysis. Modules are identified along the capabilities dimension (x axis) of the AI=MC$^2$ taxonomy. For better readability, only top-level methods and capabilities are displayed here.

### 3.2. Step 2: Methods Analysis

Let us assume that during the capabilities analysis, three distinct modules have been identified. Module A is identified by *sense/external/hear*. Module B is identified by *process/conceptual/classify*. And module C is identified by *communicate/without feedback/unidirectional.* During methods analysis, suitable AI methods for each module will be identified. As a result of this step, a method is determined for each module that will be used to implememt the related capability. In particular, this completes the task of designing a module from an architectural perspective.

For each identified module, we would in this step switch the point of view to the methods dimension, consisting of the top-level methods traditional AI, symbolic AI, machine learning and hybrid learning. An initial design decision would be whether a data-driven or a knowledge-based approach is to be chosen. For modules A and C, for example, one would typically choose a data-driven approach from the area of machine learning (Fig. 3). In order to actually implement module A (hear) the corrresponding second-level methods (here: reinforcement learning, supervised learning, semi-supervised learning, unsupervised learning, and adversarial learning) will be reviewed. Due to many existing references architectures, we might end up implementing module A by supervised learning and module C by adversarial learning.

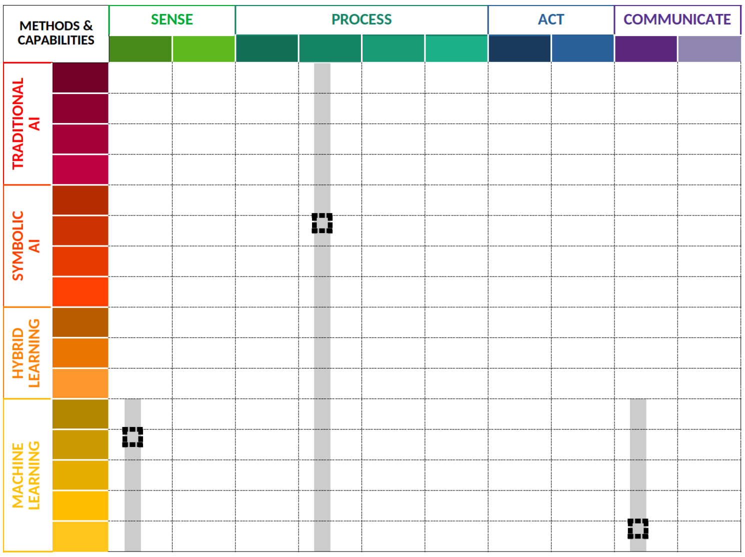


**Figure 3:** Methods analysis. For each identified module, available methods (in grey) are reviewed and an optimal method (black dots) is selected for each module.

### 3.3. Step 3: Pathway and Interface Definition

Now we have arrived at a point in the design process where all relevant modules have been identified and specified. So far undefined, however, is how these modules will interact with each other in the actual system. Therefore, the ordering, interaction, and related interfaces will be defined for all modules in this third step of the process. This may be seen as the actual design with modules and will be guided by the overall intended functionality.

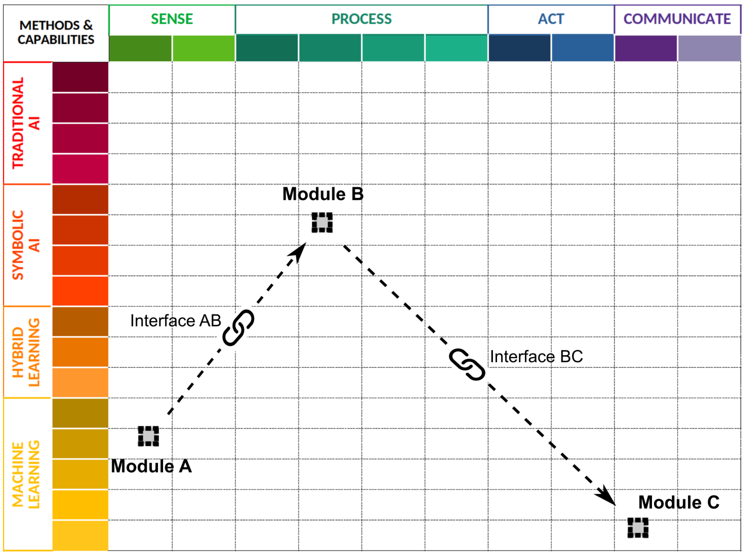


**Figure 4:** Pathway and interface definition. Connections between all specified modules are reviewed and appropriate interfaces between connected modules are defined.

In our example of designing an audio chatbot, for example, we could end up defining a connection sequence of module A $\rightarrow$ module B $\rightarrow$ module C. More specifically, we would further define the corresponding interfaces AB and BC in a way that module A expects an audio input (speech) and delivers written text (derived from audio input) as input for module B, which in turn will provide written text as input for module C, which will then generate speech (audio) from this text. All together, our toy audio chatbot specified by these three modules will then be able to accept audio input, proceed it according to the intended functionality, and finally return an audio message as answer to the user. Based on this architecural specification, development teams can take up the actual implementation process – or reuse existing modules.

## 4. Conclusions

We have introduced the AI=MC$^2$ taxonomy with a focus on its use for designing modular hybrid AI systems in a structured, systematic and efficient way. We have demonstrated that the multi-level granularity of this taxonomy regarding both methods and capabilities of AI systems provides an ideal basis for the design of modular AI systems. Moreover, we have defined and exercised a three-step design procedure based on this taxonomy. In conclusion, we have introduced a novel and efficient design strategy for modular hybrid AI.Given that this is only an initial overview of the concept, more practical specifications of the design procedure will, of course, be necessary. In future work, we will therefore further elaborate the described procedure into more detailed guidelines and practices for the design of AI systems.

## References


[1] J. K. Gershenson, G. Prasad, Y. Zhang, Product modularity: definitions and benefits, Journal of Engineering design 14 (2003) 295–313.

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

[3] A. Albers, H. Scherer, N. Bursac, G. Rachenkova, Model based systems engineering in construction kit development–two case studies, Procedia CIRP 36 (2015) 129–134.

[4] L. Manenti, The operational concepts in the vitruvian system of design, Revista Archai (2019) e02605–e02605.

[5] W. Voigt, Vitruv der moderne: Ernst neufert, in: W. Prigge (Ed.), Ernst Neufert. Normierte Baukultur im 20. Jahrhundert, Campus Verlag, 1999, pp. 20–34.

[6] B. Flyvbjerg, Make megaprojects more modular, Harvard Business Review (2021) 58–63.

[7] P. Wegner, Concepts and paradigms of object-oriented programming, ACM Sigplan Oops Messenger 1 (1990) 7–87.

[8] E. Gamma, R. Johnson, R. Helm, R. E. Johnson, J. Vlissides, Design patterns: elements of reusable object-oriented software, Pearson Deutschland GmbH, 1995.

[9] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Advances in neural information processing systems 33 (2020) 1877–1901.

[10] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, High-resolution image synthesis with latent diffusion models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10684–10695.

[11] A. Martin, K. Hinkelmann, A. Gerber, D. Lenat, F. van Harmelen, P. Clark, Preface: Preface: Combining machine learning with knowledge engineering, in: AAAI Spring Symposium on Combining Machine Learning with Knowledge Engineering, AAAI-MAKE, 2019.

[12] S. Wermter, R. Sun, An overview of hybrid neural systems, in: International Workshop on Hybrid Neural Systems, Springer, 2000, pp. 1–13.

[13] D. Yu, B. Yang, D. Liu, H. Wang, A survey on neural-symbolic systems, arXiv preprint arXiv:2111.08164 (2021).

[14] N. Slonim, Y. Bilu, C. Alzate, R. Bar-Haim, B. Bogin, F. Bonin, L. Choshen, E. Cohen-Karlik, L. Dankin, L. Edelstein, et al., An autonomous debating system, Nature 591 (2021) 379–384.

[15] M. van Bekkum, M. de Boer, F. van Harmelen, A. Meyer-Vitali, A. t. Teije, Modular design patterns for hybrid learning and reasoning systems: a taxonomy, patterns and use cases, Applied Intelligence 51 (2021) 6528–6546.

[16] T. Schmid, W. Hildesheim, T. Holoyad, K. Schumacher, The AI methods, capabilities and criticality grid, KI - Künstliche Intelligenz 35 (2021) 425–440.

[17] DIN (Ed.), German Standardization Roadmap on Artificial Intelligence, 2nd ed., 2022.

[18] T. Schmid, W. Hildesheim, T. Holoyad, K. Schumacher, Managing and Understanding Artificial Intelligence Solutions, 1st ed., Beuth, 2022.

[19] T. Schmid, W. Hildesheim, T. Holoyad, K. Schumacher, Künstliche Intelligenz managen und verstehen, 1st ed., Beuth, 2023.

[20] W.-Y. Liang, C.-C. Huang, The agent-based collaboration information system of product development, International Journal of Information Management 22 (2002) 211–224.