Model-Driven System Performance Engineering

ESI programs bring together industrial and academic partners, literally using industry as a lab. The system performance program targets the development of industrially mature, domain-specific model-driven system performance engineering methods based on the latest academic results.

Opportunity

System performance often brings the **competitive advantage** for high-tech **cyber-physical systems** like semiconductor equipment, production printers, analytical instruments, and medical equipment. To meet market demands for product quality, product customization, and a low total cost of ownership, systems need to meet ever more ambitious performance targets relating to system productivity. Performance is a cross-cutting system-level concern, with intricate relations to other system-level concerns like product quality, cost, reliability, and customizability.

Target group	High Tech industry
Contact	Bram van der Sanden Twan Basten
Address	High Tech Campus 25 5656 AE Eindhoven The Netherlands
Phone	+31 (0)88 866 54 20
E-mail	bram.vandersanden@tno.nl twan.basten@tno.nl
Information	www.esi.nl

Powered by industry, academia and TNO

System performance: the amount of useful work done by a system - measured in production speed of products of a predefined quality.

Designing for performance implies that system performance is a first-class citizen integrally taken into account during the entire system life cycle. **Model-driven system performance engineering (MD-SysPE)** is essential to make the right design decisions during early stages of system development and to optimize performance during system operation. Early insight in system performance **improves time-to-quality** by requiring less rework in later stages of development. Designing for performance **improves the cost-performance ratio** of the final product by minimizing system over-dimensioning. It also **enables a wider range of system variants and operating conditions** by considering the system variability and context during design and operation.

Industrial challenges

Eight industrial challenges motivate the importance of MD-SysPE:

1. **Performance analysis during early development**
2. **System-level modeling and reasoning**
3. **System performance (de)composition**
4. **System variability**
5. **Runtime management**
6. **Methodology: languages, techniques, methods, tools**
7. **Traceability**
8. **Data**

Performance problems often only materialize late in the system lifecycle, typically during integration or system operation. To prevent costly rework and avoid performance degradations, **performance analysis** is needed **during early system development. System-level modeling and reasoning** at the right abstraction level needs to support reasoning on performance and the trade-offs with other KPIs. Given a system architecture and system-level performance requirements, an important challenge is how to **decompose system performance** to component-level budgets and identify the impact on performance of different possible decompositions.

Systems are typically no longer designed individually. They evolve over time and are part of product families. It is not feasible to develop, evaluate, and test variants individually. It is a challenge to deal with **system variability** in relation to performance, to create the right modular platform with re-usable components, and to assess how design choices on shared components impact specific system variants. **Runtime management** is needed to recognize and react to both anticipated and unanticipated operating conditions, to ensure optimal system performance under varying circumstances.

Industrially-usable methodologies consisting of coherent sets of **languages, techniques, methods, and tools** are needed to specify, model and analyze performance, covering all aspects of all disciplines relevant for system performance, including for instance software, mechanics, and electronics. **Traceability** of performance relations across levels of abstraction, across system components, and across disciplines is needed to reason on the impact of design decisions on performance and to diagnose the root-cause of performance problems. **Data** plays a crucial role in both the design and diagnostics of systems. A key challenge lies in getting the right data with proper instrumentation of the system, while ensuring minimal impact on system performance.

Focus areas

ESI advocates MD-SysPE at system level throughout the whole system life cycle to address the identified industrial challenges. We develop domain-specific conceptual modeling techniques to capture all relevant aspects across all disciplines in a particular domain. We link performance models to analysis, synthesis, scheduling, and control techniques that enable automated reasoning about the impact of design choices on system performance, constructive design-space exploration, and on-line performance optimization. We collect data during system operation to provide feedback on the design.

Our MD-SysPE approach identifies five focus areas that together cover the full system life cycle and one cross-cutting focus area on methods and tool support:

1. **Performance architecting**
2. **Model-driven design-space exploration**
3. **Performance modeling and analysis**
4. **Scheduling and supervisory control**
5. **Data-driven analysis and design** (including data collection and model learning)
6. **Tool-supported MD-SysPE methods**

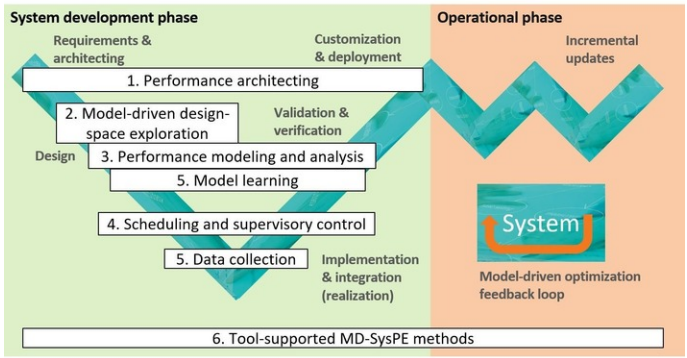

Figure 1: Focus areas positioned on a system development process

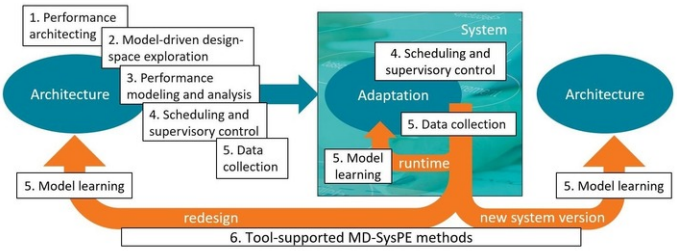

Figure 2: System life cycle with a positioning of the focus areas

Figure 1 shows a positioning of the focus areas in a typical V-model development process. While the system is in operation, incremental development iterations may be performed to update the system or to adapt it to specific operating conditions, as illustrated by the small V-development iterations at the right. Figure 2 positions the focus areas in the system life cycle, emphasizing the feedback cycles from system operation to improve system performance. Together, the figures show how the focus areas cover all aspects of system development and system operation that are relevant for MD-SysPE.

Designing for performance starts with **performance architecting**, to determine the performance aspects that need to be considered during the system life cycle, to balance those with other system-level concerns, and to ensure that the system architecture fits with the performance requirements.

In the early design phases, **model-driven design-space exploration** (DSE) is performed to explore trade-offs and find optimal designs within the given system architecture. Exploration is model-driven, at a high abstraction level, following pre-defined patterns to systematically cope with complexity. **Performance modeling and analysis** techniques are used to capture and analyze the performance of specific system configurations. Techniques are targeted to the type of systems and performance requirements at hand, ranging from analytical modeling and reasoning about performance bounds to discrete-event simulation and stochastic reasoning about expected performance.

Scheduling and supervisory control techniques are essential to achieve the required performance during system operation. Schedulers and control strategies need to be designed during system development. Scheduling and control may then be optimized during system operation when the full operating conditions and all system inputs are known. Such on-line computations need to be done within strict time budgets and with the often limited processing resources available during system operation.

Accurate models are essential for all the activities mentioned. **Data-driven analysis and design** techniques enable model validation, model calibration, and model learning. Operational data can be used for monitoring performance targets during system operation, for diagnosing unexpected performance degradations, for development of system updates, and for system (re-)design. Selecting the right data to be collected and lightweight, non-intrusive system instrumentation are essential.

The five focus areas discussed so far provide the basic techniques needed for MD-SysPE. These techniques need to be integrated in mature industrially-usable **tool-supported MD-SysPE methods**, that cover all relevant disciplines and the full system life cycle.

Best practices

For the first five focus areas, we identified and developed best practices to address the challenges mentioned earlier.

Performance architecting

In architecting, it is important to **include a performance view in reference architectures** as architectural choices can have a significant impact on achievable system performance. Architecting commonly distinguishes between *artifacts, domain models,* and *aspect models*. Artifacts, such as documentation, code, and system data, describe the current system. Domain models generalize the essential domain concepts and their inter-relations beyond the scope of specific systems to the domain at hand, covering all relevant disciplines. The domain models link to aspect models that enable analysis of particular aspects of design alternatives. For performance architecting, **domain and aspect models make performance aspects explicit**.

Platform-based design and **budget-based design** form a basis for first-order system decomposition. Platform-based design targets the development of re-usable components, subsystems, and technology (comprehensively referred to as platforms). Budget-based design aims to budget critical aspects, including performance, and resources in a design. When integrated in a model-driven design flow, e.g., through virtual prototypes, platforms and budgeting help to speed up the development process, to better evaluate project risks, and to obtain better design trade-offs that explicitly consider performance during early design.

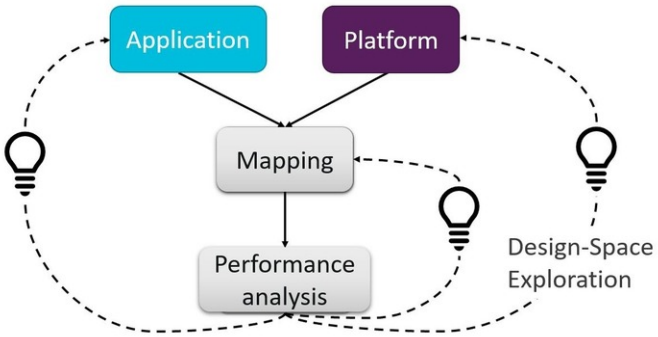

Figure 3: Y-chart paradigm

Model-driven DSE

Model-based DSE is the process of iterative **model-based prediction and validation of the performance of design alternatives**, to obtain feedback to the development process. Model-based DSE is a step towards *model-driven* DSE, in which models drive the development process and are the single source of truth. Model-driven DSE uses *virtual prototypes* for performance prediction and exploration. In the development of system variants or product families, DSE follows the **predict-the-past, explore-the-future paradigm**. The past is predicted by creating aspect models of existing systems or system components, calibrated or validated with measurements. This gives prediction accuracy and builds trust. To explore the future, performance aspects of design alternatives can be analyzed with models adapted to these design alternatives.

To facilitate effective and efficient DSE, it is important to separate concerns regarding system functionality and implementation aspects. The **Y-chart paradigm** proposes to model application functionality and the implementation platform as separate elements, with an explicit mapping between them. This allows to easily vary functionality, platform resources, and mapping choices and analyze the performance impact of these choices.

Performance modeling and analysis

Performance modeling is typically done using a combination of knowledge-driven and data-driven modeling. **Knowledge-driven modeling** builds on expert knowledge of domain specialists. **Data-driven modeling** creates models through regression or model learning from data collected from prototypes, tests, or system operation. We strive for models with rigorous mathematical foundations, such as max-plus linear systems in LSAT (https://lsat.esi.nl) or timed stochastic decision processes in POOSL (https://poosl.esi.nl).

Performance analysis can be done by **discrete-event simulation** (e.g. in POOSL), to analyze specific system behaviors, by **analytical analysis** (e.g. in LSAT), to derive performance bounds, or by **model checking**, to exhaustively verify performance properties on a system model. The latter can effectively be used, for instance, on execution or model traces (Gantt charts) built from actions, events, and signals, as done for example in TRACE (https://trace.esi.nl).

Scheduling and supervisory control

Supervisory controllers and schedulers need to realize correct system operation optimizing performance in relation to other system-level concerns like product quality, reliability, or cost. Based on models of system behavior, (template code for) **schedulers and controllers can be synthesized**. Those schedulers and controllers should **optimize performance at runtime**, for varying system configurations and operating conditions. They should preferably guarantee **performance by construction**. For instance, a scheduler may guarantee a minimum productivity under varying operating conditions. Or a controller can minimize energy usage at runtime taking into account operational information on energy usage while ensuring that the expected number of missed deadlines does not exceed a threshold of 2%.

Data-driven analysis and design

With the availability of large quantities of data and computing power, data-driven modeling, analysis, scheduling, and control complement their knowledge-based counterparts. Operational data may be used to improve system performance. It is essential to collect the **right data** via **lightweight, non-intrusive system instrumentation and timing measurements**. It is important to **ensure timing accuracy** (e.g., via clock synchronization) and to consider **storage** and **bandwidth** limitations that determine whether the required data can be collected in a real-time manner or not. Operational data can be related to models through **model validation and calibration**, **model learning**, or **digital twinning, balancing knowledge-driven and data-driven design approaches**.

Benefits

MD-SysPE provides insight in system performance, supports well-founded design decisions, and enables performance optimization during system design and operation. The identified best practices **improve time-to-quality** and **cost-performance ratio** of systems and enlarge their **productive operating range**.