ORNL/TM-2023/3133

INTERSECT Architecture Specification: Use Case Design Patterns (Version 0.9)

Christian Engelmann and Suhas Somnath

September 30, 2023

Approved for public release. Distribution is unlimited.

OAK RIDGE National Laboratory

ORNL/TM-2023/3133

Laboratory Directed Research and Development Program
Self-Driven Experiments for Science/Interconnected Science Ecosystem (INTERSECT) Initiative

INTERSECT Architecture Specification: Use Case Design Patterns (Version 0.9)

Christian Engelmann and Suhas Somnath

September 30, 2023

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, TN 37831-6283
managed by
UT-Battelle LLC
for the
US DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

ACL autonomous robotic chemistry laboratory. v, vi, xv, 1, 18, 22, 36, 39, 42, 45, 62–65
AGILE Automation for Grid Interconnected-Laboratory Emulation. v, 50, 51
AI artificial intelligence. xiii, 1, 2, 5, 7, 10, 16, 19, 36, 39, 46, 62
AM additive manufacturing. v, vi, 52–54
AutoFlowS Autonomous Continuous Flow Reactor Synthesis. vi, xv, 29, 55–58
CNMS Center for Nanophase Materials Sciences. 59
CPU central processing unit. 59
DAG directed acyclic graph. 20, 41, 44
DFT density functional theory. 59
DoDAF DoD Architecture Framework. 6
DOE U. S. Department of Energy. 1
ESS energy storage system. 50
EV electric vehicle. 50
GPGPU general-purpose computing graphics processing unit. 59
GRID-C Grid Research Integration and Deployment Center. 50, 51
GUI graphical user interface. 32, 47, 48, 60
HMI human-machine interface. 33, 60
HPC high-performance computing. xiii, 7, 59
INTERSECT Self-driven Experiments for Science / Interconnected Science Ecosystem. v, xiii, 2, 3, 6
IR infra-red. 52
MD molecular dynamics. 59
NAS network attached storage. 48
OO object-oriented. 7
OODA observe, orient, decide, and act. 5, 28, 29, 31, 32, 34, 35, 37, 38
OPL Our Pattern Language. 7
ORNL Oak Ridge National Laboratory. v, 2, 3, 6, 52, 59
PE power electronics. 50, 51
PID proportional–integral–derivative. 10, 16
PV photovoltaic. 50
SME subject matter expert. 18, 53, 55, 62
SNS Spallation Neutron Source. 45, 52
SoS system of systems. 2, 6
STEM scanning transmission electron microscopy. 16, 32, 33, 59, 60
TBB Thread Building Blocks. 7

ACKNOWLEDGEMENTS

Research sponsored by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the U. S. Department of Energy.

ABSTRACT

Connecting scientific instruments and robot-controlled laboratories with computing and data resources at the edge, the Cloud or the high-performance computing (HPC) center enables autonomous experiments, self-driving laboratories, smart manufacturing, and artificial intelligence (AI)-driven design, discovery and evaluation. The Self-driven Experiments for Science / Interconnected Science Ecosystem (INTERSECT) Open Architecture enables science breakthroughs using intelligent networked systems, instruments and facilities with a federated hardware/software architecture for the laboratory of the future. It relies on a novel approach, consisting of (1) science use case design patterns, (2) a system of systems architecture, and (3) a microservice architecture. This document introduces the science use case design patterns of the INTERSECT Architecture. It describes the overall background, the involved terminology and concepts, and the pattern format and classification. It further details the 12 defined patterns and provides insight into building solutions from these patterns. The document also describes the application of these patterns in the context of several INTERSECT autonomous laboratories. The target audience are computer, computational, instrument and domain science experts working in the field of autonomous experiments.

REVISION RECORD

Version	Date	Description
0.9	09/30/2023	Internal draft and review release with the following changes: • Edits from the EuroPLoP paper review, improving readability and consistency • Improved abstract and introduction • Added autonomous additive manufacturing use case
0.8	06/30/2023	Internal draft and review release with the following changes: • Edits from the EuroPLoP paper review, improving readability and consistency • Improved graphics with color scheme • Added autonomous microscopy use case • Added autonomous robotic chemistry laboratory (ACL) use case • Added Autonomous Continuous Flow Reactor Synthesis (AutoFlowS) use case
0.7	03/31/2023	Internal draft release with the following changes: • Added Local and Distributed Multi-Experiment Workflow architectural patterns
0.6	31/12/2022	Internal draft release with the following changes: • Changed naming scheme for architectural patterns from Remote ... to Distributed ... • Improved the autonomous microscopy section
0.5	09/30/2022	Public release with the following changes: • Further clarified terminology in pattern component diagrams (using "Test") • Added section about design pattern compositions • Removed sections for INTERSECT case studies from public release due to lack of maturity
0.4	06/30/2022	Internal draft release with the following changes: • Added experiment result to terminology and pattern descriptions/diagrams • Added optional post-processing of raw experiment result to all architectural patterns • Clarified language about remote components in architectural patterns • Added legend about data/control paths to diagrams in architectural patterns • Added a section for building solutions using science use case design patterns • Added sections for INTERSECT case studies
0.3	04/04/2022	Internal draft release with the following changes: • Clarified terminology and concepts • Improved science use case design pattern introduction • Extended science use case design pattern format by clarifying existing pattern properties and adding new pattern properties • Reorganized science use case design pattern classification • Added 2 architectural patterns (local and remote experiment control) • Significantly extended the descriptions of all patterns in the catalog, added new pattern property descriptions and improved pattern component diagrams
0.2	01/06/2022	Internal draft release
0.1	12/31/2021	Initial, unpublished draft

1. INTRODUCTION

The U. S. Department of Energy (DOE)'s Artificial intelligence (AI) for Science report [23] outlines the need for intelligent systems, instruments, and facilities to enable science breakthroughs with autonomous experiments, self-driving laboratories, smart manufacturing, and AI-driven design, discovery and evaluation [29]. The DOE's Computational Facilities Research Workshop report [1] identifies intelligent systems/facilities as a challenge with enabling automation and reducing human-in-the-loop needs as a cross-cutting theme.

Autonomous experiments, self-driving laboratories and smart manufacturing employ machine-in-the-loop intelligence for decision-making. Human-in-the-loop needs are reduced by an autonomous online control that collects experiment data, analyzes it, and takes appropriate operational actions to steer an ongoing or plan a next experiment. It may be assisted by an AI that is trained online and/or offline with archived data and/or with synthetic data created by a digital twin. Analysis and decision making may also rely on rule-based approaches, causal or physics-based models, and advanced statistical methods. Human interaction for experiment planning, observation and steering is performed through appropriate human-machine interfaces.

For example, both the rate and output of traditional materials synthesis and discovery are currently too slow and too small to efficiently provide needed advances. An autonomous robotic chemistry laboratory (ACL) (Figure 1-1) can operate 24/7 with high precision to greatly accelerate materials discovery and innovation. It relies on the design of a laboratory utilizing robotic and autonomous tools for the manipulation of laboratory equipment and characterization tools. A robotic platform with three major components is used: a mobile base, a robotic arm, and software/characterization tools including integration/feedback with AI.

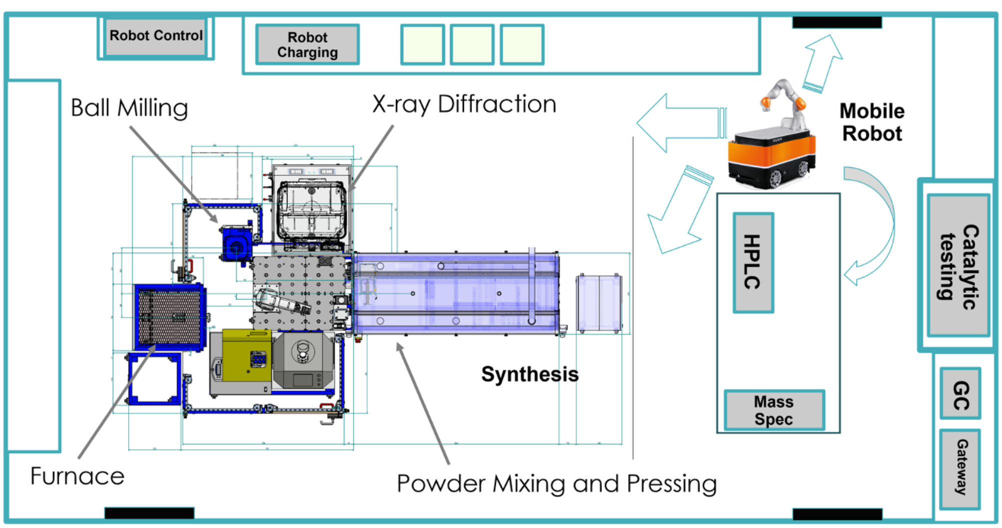


Figure 1-1. The INTERSECT autonomous robotic chemistry laboratory operates 24/7 using analysis of experimental data for the design of experiments.

A federated hardware/software ecosystem (Figure 1-2) for connecting instruments with edge and center computing resources is needed that autonomously collects, transfers, stores, processes, curates, and archives scientific data in common formats. It must be able to communicate with scientific instruments and computing and data resources for orchestration and control across administrative domains, and with humans for critical decisions and feedback. Standardized communication and programming interfaces are needed that leverage community and custom software for scientific instruments, automation, workflows and data transfer. Pluggability is required to permit quickly adaptable and deployable solutions, reuse of partial solutions for different use cases, and the use of digital twins, such as a virtual instrument, robot or experiment. This federated ecosystem needs to follow an open architecture standard to enable adoption.

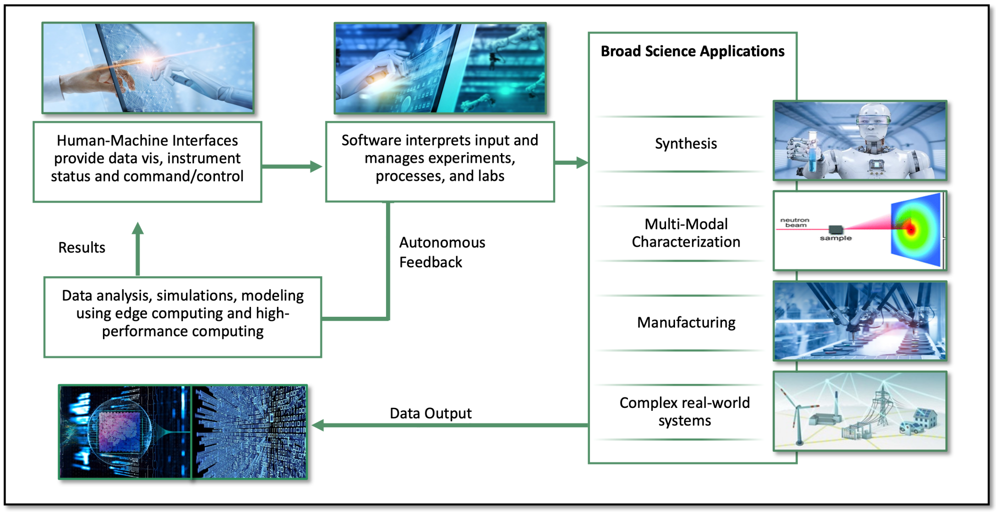


Figure 1-2. The INTERSECT ecosystem vision connects instruments with edge and center computing resources.

Oak Ridge National Laboratory (ORNL)'s Self-driven Experiments for Science / Interconnected Science Ecosystem (INTERSECT) architecture project (Figure 1-3), titled "An Open Federated Architecture for the Laboratory of the Future", creates an open federated hardware/software architecture for the laboratory of the future using a novel system of systems (SoS) and microservice architecture approach, connecting scientific instruments, robot-controlled laboratories and edge/center computing/data resources to enable autonomous experiments, self-driving laboratories, smart manufacturing, and AI-driven design, discovery and evaluation.

The project describes science use cases as design patterns that identify and abstract the involved hardware/software components and their interactions in terms of control, work and data flow. It creates a SoS architecture of the federated hardware/software ecosystem that clarifies terms, architectural elements, the interactions between them and compliance. It further designs a federated microservice architecture, mapping science use case design patterns to the SoS architecture with loosely coupled microservices, standardized interfaces and multi programming language support. The primary deliverable of this project is an INTERSECT Open Architecture Specification, containing the science use case design pattern catalog, the federated SoS architecture specification and the federated microservice architecture specification.

This document introduces the science use case design patterns of the INTERSECT Open Architecture Specification. The basic template for a science use case design pattern is defined in a loop control problem paradigm. At the moment, there are two classes of science use case design patterns, based (1) on high-level solution methods using experiment control architecture features at a very coarse granularity and (2) on more specific solution methods using hardware and software architecture features at a finer granularity. The classification scheme itself is open for extension, such as for adding new patterns for each class or new

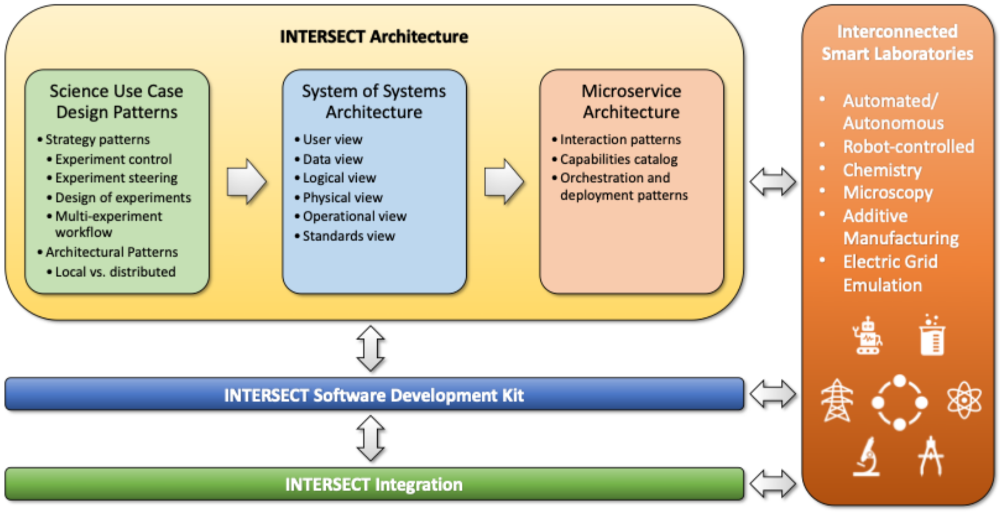


Figure 1-3. The INTERSECT Architecture project in the context of ORNL's INTERSECT Initiative.

classes entirely. For example, a new class may map the existing patterns to other workflow properties, such as (a) data-intensive, (b) time-sensitive and (c) long-term experiment campaigns. The document describes the overall background, the involved terminology and concepts, and the pattern format and classification. It further details the 12 defined patterns and provides insight into building solutions from these patterns. The document also describes the application of these patterns in the context of several INTERSECT autonomous laboratories. The target audience are computer, computational, instrument and domain science experts working in the field of autonomous experiments.

2. TERMINOLOGY AND CONCEPTS

This section briefly describes some of the used terminology and concepts in this document for clarification purposes. The descriptions are thematically grouped and ordered by dependence, i.e., their relationships. Although many of the used terminology and concepts have additional meanings in science and engineering, their descriptions are based on the context they are being used in this document. This is by no means an exhaustive list, but rather represents the core descriptions needed to understand this document.

General Terms:

- **Test:** A procedure or a method to evaluate the characteristics of a product, service, or system under specific conditions. For example, characterizing the chemical composition of a compound in a gas chromatograph.
- **Experiment:** A test under controlled conditions to demonstrate a known truth or examine the validity of a hypothesis. For example, creating a compound based on the hypothesis that it has a certain chemical composition, characterizing the chemical composition of the compound in a gas chromatograph, and analyzing the result to examine the validity of the hypothesis.
- **Multi-experiment workflow:** A set of experiments performed in serial (one after another) and/or in parallel (simultaneously). For example, a created compound is characterized with different tools, including a gas chromatograph, to examine the validity of multiple hypotheses. This may be performed by splitting the compound up and performing the experiments simultaneously (parallel), or by reusing the compound in subsequent (serial) experiments.
- **Campaign:** A scientific endeavor that may consist of one or more experiments that may take place sequentially or in parallel to answer a broader overarching scientific question. For example, performing multiple experiments involving a gas chromatograph in which different compounds are created and characterized to find an optimal compound for a specific practical application.

Experiment and Multi-experiment Workflow Data:

- **Experiment plan:** A list of actions that need to be executed while running an experiment.
- **Experiment design plan:** An initial experiment plan and a plan for creating new experiment plans based on experiment results.
- **Multi-experiment workflow plan:** A list of actions that need to be executed while running a multiple experiments in a workflow, i.e., a set of experiments in serial and/or parallel. Each experiment in this workflow still has its own experiment plan.
- **Experiment result:** The data collected from sensors before, during and/or after running an experiment.

Operational Experiment Properties:

- **Automated:** Executing an existing experiment or multi-experiment workflow plan, by performing its list of actions, without external or human intervention that can unnecessarily hold up execution.
- **Autonomous:** Creating a new or modifying an existing experiment or multi-experiment workflow plan and executing it, by performing its list of actions, without external or human intervention that can unnecessarily hold up execution.
- **Self-driving:** Synonymous with autonomous operation.

Experiment Devices:

- **Sensor:** A device for measuring something before, during and/or after running an experiment.
- **Actuator:** A device for moving or controlling something before, during and/or after running an experiment.
- **Instrument:** A device containing sensors and potentially actuators.
- **Robot:** An automated or autonomous device containing actuators and potentially sensors.
- **Laboratory:** A room or building equipped with experiment devices, such as sensors, actuators, instruments, and robots.

Experiment and Workflow Control:

- **Loop control:** The devices and functions necessary to automatically or autonomously perform an experiment or a multi-experiment workflow.
- **Open loop control:** A loop control without feedback, except to monitor the experiment(s) for safety reasons.
- **Closed loop control:** A loop control with feedback, such as to monitor experiment(s) progress or result and to adapt experiment or multi-experiment workflow plans.
- **Observe, orient, decide, and act (OODA) loop control:** A closed loop control with 4 distinct components: (1) *Observe* the evolving situation, (2) *Orient* the observed information for decision making, (3) *Decide* on appropriate actions, and (4) *Act* on the made decisions [25].
- **Experiment controller:** A component that executes an experiment plan by performing its list of actions and collecting any feedback.
- **Experiment planner:** A component that creates an experiment plan based on an experiment design plan and experiment results.
- **Multi-experiment workflow controller:** A component that executes a multi-experiment workflow plan by performing its list of actions and collecting any feedback.

Other Terms:

- **Smart manufacturing:** Computer-integrated manufacturing with high levels of adaptability and rapid design changes, treating the manufacturing process as series of experiments that improve the product through feedback.

- **AI-driven design, discovery and evaluation:** The use of AI technology in product design, scientific discovery, or product evaluation/testing.

3. DESIGN PATTERNS FOR SCIENCE USE CASES

The primary deliverable of ORNL's INTERSECT architecture project, titled "An Open Federated Architecture for the Laboratory of the Future", is an INTERSECT Open Architecture Specification, containing a use case design pattern catalog, a SoS architecture specification and a microservice architecture specification. It essentially offers different viewpoints of the INTERSECT Open Architecture, similar to the DoD Architecture Framework (DoDAF) [26] approach with its (1) operational scenarios; (2) composition, interconnectivity and context; (3) services and their capabilities; (4) policies, standards and guidance; and (5) capability.

The SoS architecture specification primarily follows the DoDAF approach and provides operational, data, logical and physical views of the architecture, among others. The use case design pattern catalog contains elements of the use case view, but is a separate document, as it follows the design pattern approach. Similarly, the microservice architecture specification contains elements of the service view, but follows the microservice architecture approach. The microservice architecture maps use case design patterns to the SoS architecture with loosely coupled microservices and standardized interfaces.

3.1 INTRODUCTION TO DESIGN PATTERNS

A design pattern is a description of a generalized solution to a recurring problem within a well-defined context. Design patterns are often created from best practices and contain the essential elements of the problems they tackle and their corresponding solutions. They offer a template on how to solve a specific problem that may apply to different situations. They may also describe different solution alternatives to a specific problem.

The concept of design patterns originates in civil architecture and engineering. Design patterns captured the detailed designs of towns and neighborhoods, houses, gardens and rooms with the goal of designing functional and aesthetically beautiful living spaces and structures. They identify and catalog solutions to recurrent problems encountered during the process of building and planning. Each pattern describes a problem that occurs repeatedly in our environment and the core of the solution to that problem in such a way that it may be used a million times over, without ever doing it the same way twice [2].

In general, a design pattern identifies the key aspects of a solution and creates an abstract description that makes it useful in the creation of a reusable design element. Patterns don't describe a concrete design or an implementation - they are intended to be templates that may be applied by a designer in various contexts and modified to suit the problem at hand. Patterns are also free from constraints of detail associated with the level of system abstraction at which the solution is implemented. Patterns also describe the design decisions that must be made when applying a certain solution. This enables a designer to reason about the impact of the design decisions on a system's flexibility or scalability as well as consider implementation issues. Design patterns must address a specific problem at hand, and yet must be general enough to remain relevant to future requirements of systems.

In the domain of software design, patterns were introduced in an effort to create reusable solutions in the design of software and bring discipline to the art of programming. The intent of software design patterns isn't to provide a finished design that may be transformed directly into code; rather, design patterns are used to enhance the software development process by providing proven development paradigms. With the use of design patterns, there is sufficient flexibility for software developers to adapt their implementation to accommodate any constraints, or issues that may be unique to specific programming paradigms, or the target platform for the software. Related to design patterns, the concept of algorithmic skeletons was introduced [6] and further refined [7].

In the context of object-oriented (OO) programming, design patterns provide a catalog of methods for defining class interfaces and inheritance hierarchies, and establish key relationships among the classes [11]. In many object-oriented systems, reusable patterns of class relationships and communications between objects are used to create flexible, elegant, and ultimately reusable software design. There are three categories of OO patterns: (i) **creational** patterns for ways to do instantiation of objects, (ii) **structural** patterns concerned with class and object composition, and (iii) **behavioral** patterns for communication between objects. Patterns have also been defined in the design of software architectures [4] to capture repeatedly used methodologies in software engineering practice. Pattern systems have also been developed for cataloging concurrent and networked object-oriented environments [22], resource management [18], and distributed software systems [5].

In the pursuit of quality and scalable parallel software, patterns for programming paradigms were developed [19] as well as a pattern language, called Our Pattern Language (OPL) [17]. These parallel patterns are used as means to systematically describe parallel computation and communication when architecting parallel software. In an effort to enable a more structured approach to designing and implementing parallel applications, particularly for many-core processors, a catalog of parallel patterns enables programmers to compose parallel algorithms, which may be easily implemented using various programming interfaces such as OpenMP, OpenCL, Cilk Plus, ArBB, Thread Building Blocks (TBB) [20]. For the design of parallel algorithms, deterministic patterns support the development of systems that automatically avoid unsafe race conditions and deadlock [21].

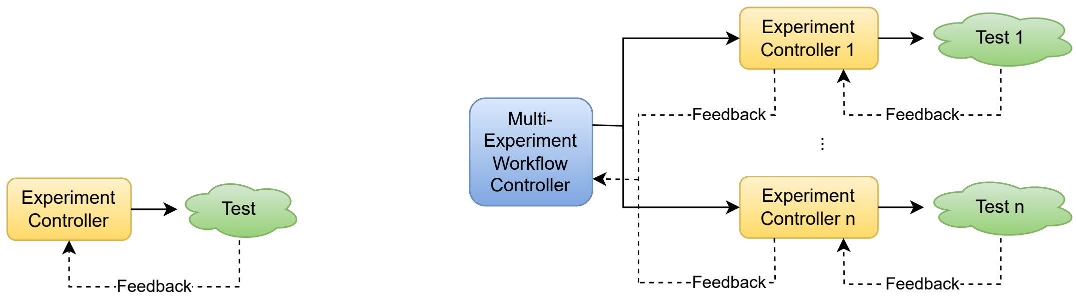
Design patterns have been identified in a variety of other domains for codifying the best-known solutions to common problems, including natural language processing [24], user interface design [3], web design [9], visualization [13], software security [8] and high-performance computing (HPC) resilience [15, 14]. Patterns have also been defined for enterprise applications that involve data processing in support or automation of business processes [10] in order to bring structure to the construction of enterprise application architectures. In each of these domains of design, patterns capture the essence of effective solutions in a succinct form that may be easily applied in similar form to other contexts and problems.

Execution patterns, not design patterns, for workflows in general describe the functionality of a workflow [27], such as execution graphs, decision points and synchronization points. Common motifs in scientific workflows [12] start making the connection between the functionality of a workflow and certain common execution patterns, such as data movement and data analysis steps. Similar workflows execution patterns, not design patterns, have been recently proposed for instrument science [28].

3.2 ANATOMY OF A SCIENCE USE CASE DESIGN PATTERN

Reducing human-in-the-loop requirements with machine-in-the-loop capabilities by connecting scientific instruments, robot-controlled laboratories and edge/center computing/data resources to enable autonomous experiments, self-driving laboratories, smart manufacturing, and AI-driven design, discovery and evaluation is an inherent open or closed loop control problem. Therefore, the basic template for a science use case design pattern is defined in a loop control problem paradigm. The abstract science use case design pattern consists of a behavior and a set of interfaces in the context of performing a single or a set of experiments in an open or closed loop control. Such an abstract definition creates universal patterns that describe solutions free of implementation details.

Figure 3-1 shows two different loop control problems. Figure 3-1a describes a closed loop control of an experiment that performs a test with some feedback to an experiment controller running the test. Figure 3-1b describes a multi-experiment workflow with a closed loop control of multiple experiments, each with their own a closed loop control. There are a number of different loop control problems that the science use case design patterns systematize and categorize.



(a) Experiment loop control problem **(b) Multi-experiment workflow loop control problem**

Figure 3-1. Science use case anatomy as experiment loop control problem and as multi-experiment workflow loop control problem

3.3 FORMAT OF A SCIENCE USE CASE DESIGN PATTERN

Design patterns for science use cases are expressed in a written form and in a highly structured format, which permits quick identification of relevant patterns given a certain problem to be solved and easy comparison of patterns regarding their applicability and capabilities. The format for describing science use case design patterns consists of individual descriptions of pattern properties, including text, diagrams, and mathematical models. It can be extended over time by adding more pattern properties and their descriptions. Patterns are described in the traditional design pattern paradigm: from context to problem to solution to resulting context. The current science use case design pattern format is as follows:

Name: A descriptive name that distinctly identifies the pattern and enables designers to think about designs in an abstract manner and communicate their design choices to others.

Context: The preconditions under which the pattern is relevant, including a description of the system before the pattern is applied.

Problem: A description of the problem that provides insight on when it is appropriate to apply the pattern. Multiple patterns may address the same problem differently.

Forces: A description of the relevant forces and constraints, and how they interact or conflict with each other and with the intended goals and objectives.

Solution: A description of the solution that defines the abstract elements that are necessary for the composition of the design solution as well as their relationships, responsibilities, and collaborations. The specific capabilities provided by this pattern.

Resulting Context: A brief description of the post-conditions arising from the application of the pattern. There may be trade-offs between competing optimization parameters that arise due to the implementation of a solution using this pattern.

Related Patterns: The relationships between this pattern and other relevant patterns. Other patterns may be predecessor or successor patterns. This pattern may complement or enhance other patterns. There may also be dependencies between patterns to provide a complete solution.

Examples: A description of one or more examples, including their specific pattern properties, that illustrate the use of the pattern for solving concrete problems.

Known Uses: A list of known applications of the pattern in existing systems, including any practical considerations and limitations.

4. CLASSIFICATION OF SCIENCE USE CASE DESIGN PATTERNS

As explained in Section 3.1, there can be different categories, or classes, of design patterns, depending on context. A classification of patterns helps to identify groups of patterns that address similar problems in different ways or that describe solutions at different levels of granularity or from different points of view. A classification scheme codifies these relationships between patterns and enables designers to better understand individual pattern capabilities and relationships. It also further helps to understand how patterns rely on each other and can be composed to form a complete solution.

The classification of science use case design patterns is work in progress at this stage of this document. At this point, there are two classes of science use case design patterns (Figure 4-1): (1) strategic patterns that define high-level solution methods using experiment control architecture features at a very coarse granularity, and (2) architectural patterns that define more specific solution methods using hardware and software architecture features at a finer granularity. While the architectural patterns do inherit the features of certain parent strategic patterns, they also address additional problems that are not exposed at the high abstraction level of the strategic patterns.

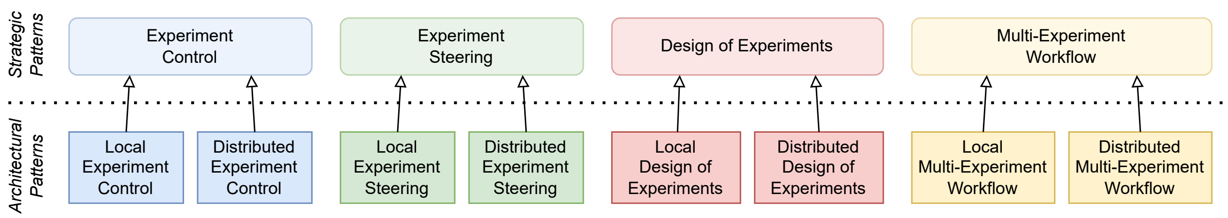


Figure 4-1. Classification of the science use case design patterns

Strategic patterns currently focus on the differences in experiment control features, such as steering of an ongoing experiment using live experimental data vs. design of the next experiment(s) using past experimental data. The key differences in features between the 4 strategic patterns are (1) no feedback, (2) feedback for the same experiment, (3) feedback for the next experiment, and (4) workflow of multiple experiments.

The primary feature currently explored by the architectural patterns is the distinction between local and remote components used by a corresponding strategic pattern, where local means that there is not a potentially significant communication delay to a component and remote means that there is a potentially significant communication delay to a component. Other architectural features may be explored in the future with different patterns.

For example, the Experiment Steering strategic pattern is used in every experiment, where live feedback of experiment data is being used to autonomously change parameters during the experiment. Known uses range from a simple proportional–integral–derivative (PID) controller to complex probabilistic approaches or domain science informed AI in the feedback loop. The Distributed Experiment Steering architectural pattern inherits all the properties of the Experiment Steering strategic pattern, but has the architectural property of potentially significant communication delay between the experiment and a remote analysis. This severely restricts real-time feedback solutions. In contrast, the Local Experiment Steering architectural pattern also inherits all the properties of the Experiment Steering strategic pattern and experiment progress is analyzed and judged locally, i.e., without significant communication delay to remote components.

This classification scheme is open for extension. New patterns may be added for each class if new strategic or architectural patterns emerge that do not fit in the existing patterns. New classes may be added if new pattern features emerge that express commonalities across workflows that are not covered by patterns. For example, a new class may map the existing patterns to data-intensive, time-sensitive and long-term experiment campaigns, which are workflow features that are orthogonal to the current pattern classes. Another new class may focus on the algorithms used in the feedback loop, such as probabilistic (e.g., Bayesian) vs. domain science based (e.g., physics informed) algorithms.

4.1 STRATEGIC PATTERNS

The science use case strategic patterns define high-level solution methods using experiment control architecture features at a very coarse granularity. Their descriptions are deliberately abstract to enable architects to reason about the overall organization of the used techniques and their implications on the full system design. The catalog of science use case design patterns defines the following strategic patterns in Section 5.1:

- *Experiment Control*: Certain predetermined actions need to be performed while running an experiment. This pattern would be used in all automated experiments that do not have feedback for steering the ongoing or designing the next experiment. Since autonomous operation requires to first figure out automation, this pattern provides a basic solution that covers most experiments performed at this point.
- *Experiment Steering*: Certain predetermined actions need to be performed while running an experiment to positively influence experiment progress. This pattern involves feedback for the ongoing experiment as an extension to Experiment Control. It offers autonomous operation and is used in experiments that require live feedback to adjust parameters.
- *Design of Experiments*: Certain predetermined actions need to be performed to run a set of similar experiments with different experiment plan parameters, depending on (prior) experiment results. This pattern makes use of either Experiment Control or Experiment Steering and additionally offers feedback between experiments, typically to define the parameters of the next experiment or next series of experiments. It is typically used in conjunction with probabilistic (e.g., Bayesian) or domain science based (e.g., physics informed) analysis of experiment results. This pattern is predominantly used in large-scale parameter studies, such as to find the optimal conditions of a chemical catalysis.
- *Multi-Experiment Workflow*: Certain predetermined actions need to be performed to run a set of experiments in serial (one after another) and/or in parallel (simultaneously). This pattern utilizes the other 3 patterns to orchestrate multiple experiments that may depend on each other. An example use case is the creation of a certain material using physical and/or chemical processes (e.g., catalysis) and the analysis of the properties of the created material in multiple experiments (e.g., spectroscopy and stress testing).

The features of these science use case strategic patterns and their relationships are compared in Table 4-1.

4.2 ARCHITECTURAL PATTERNS

Architectural patterns define more specific solution methods using hardware and software architecture features at a finer granularity. They offer more detailed descriptions, conveying different design choices for implementing strategic patterns and their abstract architectural features. Architectural patterns inherit the

Table 4-1. Feature comparison and relationships of the science use case strategic patterns

Feature	Experiment Control	Experiment Steering	Design of Experiments	Multi-Experiment Workflow
# of experiments	1	1	Multiple	Multiple
Control type	Open loop	Closed loop	Closed loop	Open loop
Operation type	Automated	Autonomous	Autonomous	Automated
Extends		Experiment Control		
Uses			Experiment Control	Experiment Control
May also use or use instead			Experiment Steering	Experiment Steering, Design of Experiments

features of their parent strategic patterns. However, they also address additional problems through specific design choices that are not exposed at the high abstraction level of the parent strategic patterns.

The architectural patterns provide abstractions for the different hardware/software architecture choices of implementing experiment control and workflow, such as using experiment-local, edge and/or center computing and data resources. The catalog of science use case design patterns defines the following architectural patterns in Section 5.2:

- *Local Experiment Control*: A local experiment controller executes an experiment. There are no remote components that could incur a significant communication delay.
- *Distributed Experiment Control*: A remote experiment controller executes an experiment, incurring a potentially significant communication delay.
- *Local Experiment Steering*: Experiment progress is analyzed and judged locally. There are no remote components that could incur a significant communication delay.
- *Distributed Experiment Steering*: Experiment progress is analyzed and optionally also judged/controlled remotely, incurring a potentially significant communication delay.
- *Local Design of Experiments*: Experiment results are analyzed and judged locally. There are no remote components that could incur a significant communication delay.
- *Distributed Design of Experiments*: Experiment results are analyzed and optionally also judged/controlled remotely, incurring a potentially significant communication delay.
- *Local Multi-Experiment Workflow*: All experiments are local. There are no remote experiments that could incur a significant communication delay.
- *Distributed Multi-Experiment Workflow*: One or more experiments are remote, incurring a potentially significant communication delay.

Table 4-2 shows the architectural patterns and their relationships to the strategic patterns.

Table 4-2. Relationships of the science use case strategic and architectural patterns

Architectural Pattern	Implements Strategic Pattern
Local Experiment Control	Experiment Control
Distributed Experiment Control	Experiment Control
Local Experiment Steering	Experiment Steering
Distributed Experiment Steering	Experiment Steering
Local Design of Experiments	Design of Experiments
Distributed Design of Experiments	Design of Experiments
Local Multi-Experiment Workflow	Multi-Experiment Workflow
Distributed Multi-Experiment Workflow	Multi-Experiment Workflow

5. CATALOG OF SCIENCE USE CASE DESIGN PATTERNS

5.1 STRATEGIC PATTERNS

5.1.1 Experiment Control

Name: Experiment Control

Context: The pattern applies to a system with the following characteristics:

- An experiment plan exists that lists the predetermined actions to be performed while running the experiment.
- Actuators may exist to allow for moving or controlling something before, during and/or after running the experiment.
- Sensors may exist to allow for measuring something before, during and/or after running the experiment.
- Instruments may exist that contain sensors and potentially actuators.
- Robots may exist that contain actuators and potentially sensors and that execute predetermined actions from the experiment plan in an automated or autonomous fashion.

Problem: Certain predetermined actions need to be performed while running an experiment.

Forces: Only pre-experiment conditions are considered in performing the predetermined actions while running an experiment. Only safety-related conditions during the experiment may be considered. Other changing conditions during the experiment or post-experiment conditions are not considered.

Solution: An experiment controller executes an experiment using a predetermined experiment plan (Figure 5-1). The plan's execution is automated, performed in an open loop control and may involve human interaction. The controller may monitor the experiment for safety reasons. The plan contains a complete description of the predetermined actions to be performed for running the experiment, including any safety-related responses.

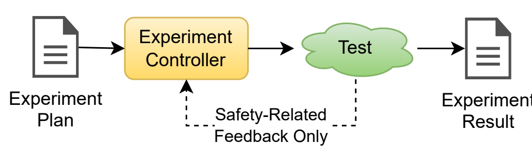


Figure 5-1. Experiment Control strategic pattern components and control/data flow

This pattern offers an open loop control with safety-related feedback on the experiment only. Experiment plan execution is automated, i.e., its list of actions is performed without external or human intervention that can unnecessarily hold up execution. Only 1 experiment is being controlled.

Resulting Context: An experiment is executed automatically using a predetermined plan.

Related Patterns: The Experiment Steering strategic pattern extends this strategic pattern with a closed loop control and feedback on experiment progress. The Multi-Experiment Workflow and Design of Experiments strategic patterns rely on this strategic pattern for automatically executing a predetermined experiment plan.

This strategic pattern is implemented by the Local Experiment Control and Distributed Experiment Control architectural patterns.

Examples: A robot-controlled chemistry laboratory science use case implements the Experiment Control strategic pattern, as a robot automates experiment execution. There is only safety-related feedback, such as to stop the robot in case of an emergency. The individual pattern components are as follows:

- The experiment plan is the sequence of predetermined steps and associated parameters necessary to run the experiment. The predetermined steps include the parameters for synthesizing the chemical compound, route navigation instructions for the robot to move the sample between the different synthesis and characterization stations, parameters for characterizing the synthesized chemical compound, and safety related feedback instructions.
- The experiment controller is a central workstation that is able to command and control the robot, synthesis equipment, analytical instruments, and any data and computing resources for analyzing the measurement data.
- The test performed in an experiment characterizes the synthesized chemical compound.
- The experiment result is a combination of the sample characterization results.

Known Uses: This strategic pattern is used in every experiment, where the predetermined experiment plan is automatically executed. The only feedback is related to safety, such as to perform an emergency stop of the experiment when a chemistry laboratory is on fire. This pattern is the starting point for any other pattern, as any feedback implemented by the Experiment Steering or Design of Experiment patterns and any orchestration of multiple experiments by the Experiment Workflow pattern require the automatic execution of a predetermined experiment plan.

5.1.2 Experiment Steering

Name: Experiment Steering

Context: The pattern applies to a system with the following characteristics:

- An experiment plan exists that lists the predetermined actions to be performed while running the experiment, including potential parameter changes based on experiment progress.
- Sensors exist to allow for measuring experiment progress.
- Actuators may exist to allow for moving or controlling something before, during and/or after running the experiment.
- Additional sensors may exist to allow for measuring something before, during and/or after running the experiment.
- Instruments may exist that contain sensors and potentially actuators.
- Robots may exist that contain actuators and potentially sensors and that execute predetermined actions from the experiment plan in an automated or autonomous fashion.

Problem: Certain predetermined actions need to be performed while running an experiment to positively influence experiment progress.

Forces: Only pre-experiment conditions and changing conditions during the experiment are considered in performing the predetermined actions while running an experiment. Post-experiment conditions are not considered.

Solution: An experiment controller executes an experiment using a predetermined experiment plan and changes the plan's parameters during execution based on experiment progress (Figure 5-2). The plan's execution is autonomous, performed in a closed loop control and may involve human interaction. The controller may monitor the experiment for safety reasons. The plan contains a complete description of the predetermined actions to be performed for running the experiment, including any safety-related responses and how to analyze and judge experiment progress and change the plan accordingly.

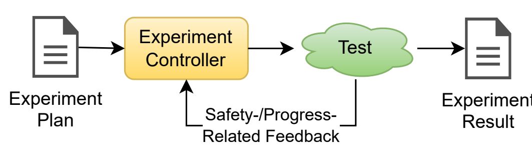


Figure 5-2. Experiment Steering strategic pattern components and control/data flow

This pattern offers a closed loop control with safety-related feedback on the experiment and feedback on experiment progress. Experiment plan execution is autonomous, i.e., its list of actions changes during execution based on feedback and is performed without external or human intervention that can unnecessarily hold up execution. Only 1 experiment is being controlled.

Resulting Context: An experiment is executed autonomously using a predetermined experiment plan, with the plan's parameters changing autonomously during the experiment based on experiment progress.

Related Patterns: This strategic pattern is an extension of the Experiment Control strategic pattern with an added closed loop control and feedback on experiment progress. The Multi-Experiment Workflow and Design of Experiments strategic patterns can be extended using this strategic pattern for autonomously executing a predetermined experiment plan, with the plan's parameters changing autonomously during experiments based on experiment progress.

This strategic pattern is implemented by the Local Experiment Steering and Distributed Experiment Steering architectural patterns.

Examples: An autonomous microscopy science use case [16] (Section 7.4) implements the Experiment Steering strategic pattern, as an ongoing scanning transmission electron microscopy (STEM) experiment is controlled by analyses of periodic experiment data. At the strategic pattern level of abstraction, the individual pattern components are as follows:

- The experiment plan contains a complete description of the predetermined actions to be performed for running the experiment, including any parameters for operating the STEM, safety-related responses and how to analyze and judge experiment progress and change the plan accordingly. The experiment plan also contains the goal of the experiment to steer it in the right direction and to stop its closed loop control upon completion.
- The experiment controller executes an experiment using a predetermined experiment plan and changes the plan's parameters during execution based on experiment progress. The plan's execution is autonomous, performed in a closed loop control and may involve human interaction.
- The test performed in an experiment determines the properties of microscopic structures.
- The experiment result is a combination of raw and analyzed STEM data and insights derived from this data.

Known Uses: This strategic pattern is used in every experiment, where live feedback of experiment data is being used to autonomously change parameters during the experiment. Known uses range from having simple PID controller to complex probabilistic approaches or domain science informed AI in the feedback loop.

5.1.3 Design of Experiments

Name: Design of Experiments

Context: The pattern applies to a system with the following characteristics:

- An experiment design plan exists that lists the predetermined actions to be performed for creating a new experiment plan based on prior experiment results.
- An initial experiment plan exists that lists the predetermined actions to be performed while running the experiment.
- Sensors exist to allow for measuring experiment results.
- Actuators may exist to allow for moving or controlling something before, during and/or after running the experiment.
- Additional sensors may exist to allow for measuring something before, during and/or after running the experiment.
- Instruments may exist that contain sensors and potentially actuators.
- Robots may exist that contain actuators and potentially sensors and that execute predetermined actions from the experiment plan in an automated or autonomous fashion.

Problem: Certain predetermined actions need to be performed to run a set of similar experiments with different experiment plan parameters, depending on experiment results.

Forces: Only pre- and post-experiment conditions are considered in performing the predetermined actions to run a set of similar experiments with different experiment plan parameters. Only safety-related conditions during the experiment may be considered. Other changing conditions during the experiments are not considered, unless the Experiment Steering strategic pattern is being used in conjunction with this strategic pattern.

Solution: An experiment controller executes each experiment using a predetermined experiment plan (Figure 5-3). The plan's execution is automated, performed in an open loop control and may involve human interaction. The controller may monitor the experiment for safety reasons. The experiment plan contains a complete description of the predetermined actions to be performed for running the experiment, including any safety-related responses. An experiment planner creates the experiment plan, based on an experiment design plan and prior experiment results (if any). The experiment plan change is autonomous, performed in a closed loop control and may involve human interaction. The experiment design plan contains an initial experiment plan and a plan for creating new experiment plans based on experiment results, including how to analyze and judge experiment results and change the plan accordingly.

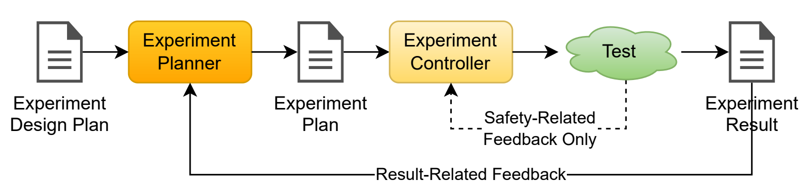


Figure 5-3. Design of Experiments strategic pattern components and control/data flow

This pattern offers an open loop control with safety-related feedback on the experiment and a separate closed loop control with feedback on experiment results. Experiment plan execution is automated within the open loop control, i.e., its list of actions is performed without external or human intervention that can unnecessarily hold up execution. Experiment design plan execution is autonomous, i.e., it creates a new experiment plan after each experiment based on experiment results and is performed without external or human intervention that can unnecessarily hold up execution. A set of similar experiments with different experiment plan parameters is controlled.

Resulting Context: An experiment is executed autonomously with different experiment plan parameters using a predetermined experiment plan, with the plan's parameters changing autonomously between experiments based on experiment results.

Related Patterns: This strategic pattern relies on the Experiment Control strategic pattern for automatically executing a predetermined experiment plan. This strategic pattern can be extended using the Experiment Steering strategic pattern (instead of the Experiment Control strategic pattern) for autonomously executing a predetermined experiment plan, with the plan's parameters changing autonomously during experiments based on experiment progress.

This strategic pattern is implemented by the Local Design of Experiments and Distributed Design of Experiments architectural patterns.

Examples: An ACL science use case (Section 7.5) implements the Design of Experiments strategic pattern, as a robot automates experiment execution and the software/characterization tools in the feedback loop to plan the experiments to be performed. At the strategic pattern level of abstraction, the individual pattern components are as follows:

- The experiment design plan describes the goal, which is the desired chemical compound, and the logic necessary to craft subsequent experiments towards achieving the goal.
- The experiment planner is the subject matter expert (SME) that may be substituted by a machine learning or deep learning model for autonomous operation to decide on the next experiment plan, given the results from past experiments.
- The experiment plan is the sequence of predetermined steps and associated parameters necessary to run the experiment. The predetermined steps include the parameters for synthesizing the chemical compound, route navigation instructions for the robots to move the sample between the different synthesis and characterization stations, parameters for characterizing the synthesized chemical compound, and safety related feedback instructions.
- The test performed in an experiment characterizes the synthesized chemical compound.
- The experiment controller is a central workstation that is able to command and control the robots, synthesis equipment, analytical instruments, and any data and computing resources for analyzing the measurement data.
- The experiment result is a combination of the sample characterization results.

The experiment is a complex sequence of steps involving multiple instruments, actuators, sensors, etc. Thus, the experiment itself could be considered a Multi-Experiment Workflow strategic pattern using a sequence of Experiment Control strategic patterns. Examples of steps that constitute the Multi-Experiment Workflow strategic include the synthesis step and each of the individual characterization steps, such as the gas chromatography, high performance liquid chromatography, and X-ray microscopy. Some of these steps could potentially be performed in parallel if the sample were broken down into pieces such that the pieces could be analyzed by the characterization instruments in parallel.

Known Uses: This strategic pattern is used in every experiment, where feedback of experiment results is being used to autonomously change the parameters of the next experiment(s). Known uses range from having simple linear or random parameter scan to complex probabilistic approaches (e.g., Bayesian design of experiments) or domain science informed AI (e.g., physics-informed design of experiments) in the feedback loop.

5.1.4 Multi-Experiment Workflow

Name: Multi-Experiment Workflow

Context: The pattern applies to a system with the following characteristics:

- A multi-experiment workflow plan exists that lists the predetermined actions to be performed for executing each experiment plan.
- An experiment plan exists for each experiment that lists the predetermined actions to be performed while running the experiment.
- Actuators may exist to allow for moving or controlling something before, during and/or after running the experiment.
- Sensors may exist to allow for measuring something before, during and/or after running the experiment.
- Instruments may exist that contain sensors and potentially actuators.
- Robots may exist that contain actuators and potentially sensors and that execute predetermined actions from the experiment plan in an automated or autonomous fashion.

Problem: Certain predetermined actions need to be performed to run a set of experiments in serial (one after another) and/or in parallel (simultaneously).

Forces: Only pre- and post-experiment conditions are considered in performing the predetermined actions to run the set of experiments in serial and/or parallel. Safety-related conditions during the experiments may be considered. Only pre-experiment conditions are considered in performing the predetermined actions while running each experiment individually, unless the Experiment Steering strategic pattern or the Design of Experiments strategic pattern are being used for some or all experiments. If the Experiment Steering strategic pattern is being used for a particular experiment, then changing conditions during this experiment are considered in performing the predetermined actions while running it. If the Design of Experiments strategic pattern is being used for a particular experiment, then post-experiment conditions are considered in performing the predetermined actions to run it with different experiment plan parameters.

Solution: A multi-experiment workflow controller orchestrates the execution of the experiments using a predetermined multi-experiment workflow plan (Figure 5-4). The multi-experiment workflow plan's execution is automated, performed in an open loop control and may involve human interaction. The multi-experiment workflow controller may monitor one or more experiment controllers for dependency reasons. The multi-experiment workflow plan contains a complete description of the predetermined actions to be performed for orchestrating the execution of the experiments including any dependency-related responses.

Multiple experiment controllers execute their experiments using their predetermined experiment plan. Each plan's execution is automated, performed in an open loop control and may involve human interaction. Each experiment controller may monitor the experiment for safety reasons. Each experiment plan contains a complete description of the predetermined actions to be performed for running its experiment, including any safety-related responses.

Some experiments may be executed in parallel, as they do not depend on each other, while other experiment may be executed in serial due to dependencies. The orchestration of the execution follows a directed acyclic graph (DAG) with the experiments as vertices and the edges as dependencies (Figure 5-5). A dependency between experiments may arise when one experiment needs the result of another.

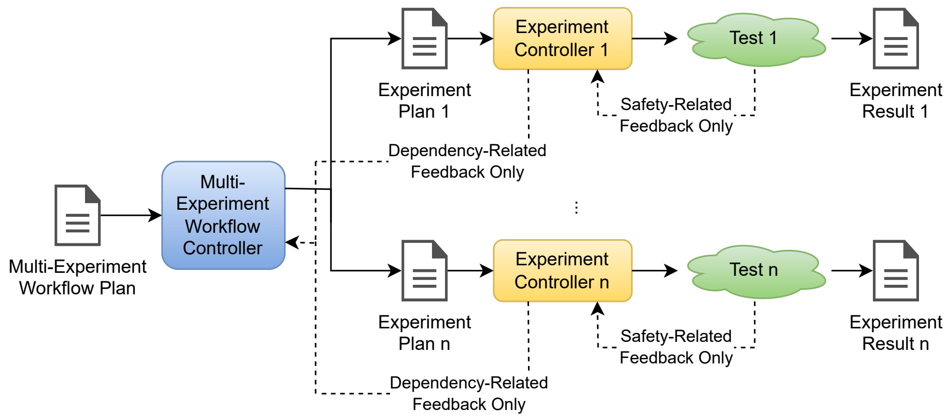


Figure 5-4. Multi-Experiment Workflow strategic pattern components and control/data flow (using multiple experiments with Experiment Control strategic patterns in this example)

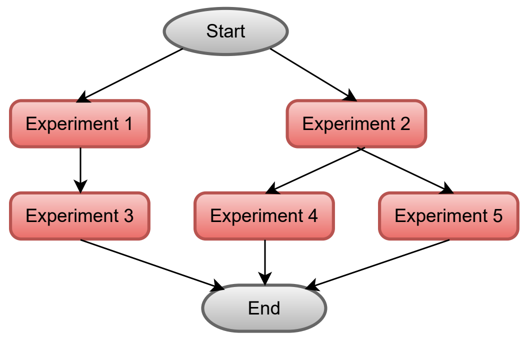


Figure 5-5. Example of a Multi-Experiment Workflow strategic pattern directed acyclic graph

This pattern offers an open loop control with safety-related feedback on each experiment and a separate loop control with safety-related feedback for each experiment. Experiment plan execution is automated within the open loop control for each experiment, i.e., its list of actions is performed without external or human intervention that can unnecessarily hold up execution. Multi-experiment workflow plan execution is automated within the open loop control for all experiments, i.e., its list of actions is performed without external or human intervention that can unnecessarily hold up execution. A set of serial and/or parallel experiments is controlled.

Resulting Context: Experiments are executed automatically in serial and/or parallel using a predetermined plan.

Related Patterns: This strategic pattern relies on the Experiment Control strategic pattern for automatically executing each predetermined experiment plan. This strategic pattern can be extended using the Experiment Steering strategic pattern (instead of the Experiment Control strategic pattern) for autonomously executing some or all predetermined experiment plans, with each plan's parameters changing autonomously during experiments based on progress. This strategic pattern can also be extended using the Design of Experiments strategic pattern for autonomously executing some or all predetermined experiment plans, with each plan's parameters changing autonomously between experiments based on results. The Experiment Control, Experiment Steering and Design of Experiments strategic patterns can be used together in conjunction with this strategic pattern, individually for each experiment of the multi-experiment workflow. However, the Experiment Control and Experiment Steering strategic patterns are mutually exclusive for the same experiment, as the Experiment Steering strategic pattern extends the Experiment Control strategic pattern.

Examples: In the ACL science use case (Section 7.5), the experiment is a complex sequence of steps involving multiple instruments, actuators, sensors, etc. Thus, the experiment itself could be considered a Multi-Experiment Workflow strategic pattern using a sequence of Experiment Control strategic patterns. Examples of steps that constitute the Multi-Experiment Workflow strategic include the synthesis step and each of the individual characterization steps, such as the gas chromatography, high performance liquid chromatography, and X-ray microscopy. Some of these steps could potentially be performed in parallel if the sample were broken down into pieces such that the pieces could be analyzed be the characterization instruments in parallel.

Known Uses: This strategic pattern is used every time a set of experiments are performed. Very common examples are (1) a set of parallel experiments that investigate the same physical sample that is getting split up beforehand, or (2) a set of serial experiments that investigate the same physical sample that is getting moved from one experiment to the next. Each of these experiments investigates different properties, where the overall combination of the experiment results may be part of a bigger experiment that encompasses them.

5.2 ARCHITECTURAL PATTERNS

5.2.1 Local Experiment Control

Name: Local Experiment Control

Context: The pattern applies to a system with the following characteristics:

- An experiment plan exists that lists the predetermined actions to be performed while running the experiment.
- A local experiment controller exists that executes the predetermined actions to be performed while running the experiment.
- Actuators may exist to allow for moving or controlling something before, during and/or after running the experiment.
- Sensors may exist to allow for measuring something before, during and/or after running the experiment.
- Instruments may exist that contain sensors and potentially actuators.
- Robots may exist that contain actuators and potentially sensors and that execute predetermined actions from the experiment plan in an automated or autonomous fashion.
- A component may exist that post-processes raw experiment data, such as to identify features.

Problem: Certain predetermined actions need to be performed while running an experiment. A local experiment controller executes an experiment. There are no remote components that could incur a significant communication delay.

Forces: Only pre-experiment conditions are considered in performing the predetermined actions while running an experiment. Only safety-related conditions during the experiment may be considered. Other changing conditions during the experiment or post-experiment conditions are not considered.

Experiment actions are controlled without significant communication delay to remote components. Proper control capability must be present locally to be able to act in time.

Solution: The is pattern implements the Experiment Control strategic pattern. A local experiment controller executes an experiment using a predetermined experiment plan (Figure 5-6). The plan's execution is automated, performed in an open loop control and may involve human interaction. The controller may monitor the experiment for safety reasons. The plan contains a complete description of the predetermined actions to be performed for running the experiment, including any safety-related responses. Raw experiment data may be post-processed by an optional component, such as to identify features.

As all components are local, a shared storage device may be used between them for sensor and controller data. Control messages between these components orchestrate the control flow.

This pattern offers an open loop control with safety-related feedback on the experiment only. Experiment plan execution is automated, i.e., its list of actions is performed without external or human intervention that can unnecessarily hold up execution. Only 1 experiment is being controlled. There is no significant communication delay to remote components in the open loop control, as the experiment controller is local.

Resulting Context: An experiment is executed automatically using a predetermined plan executed by a local experiment controller, i.e., without significant communication delay to remote components.

Related Patterns: This architectural pattern implements the Experiment Control strategic pattern.

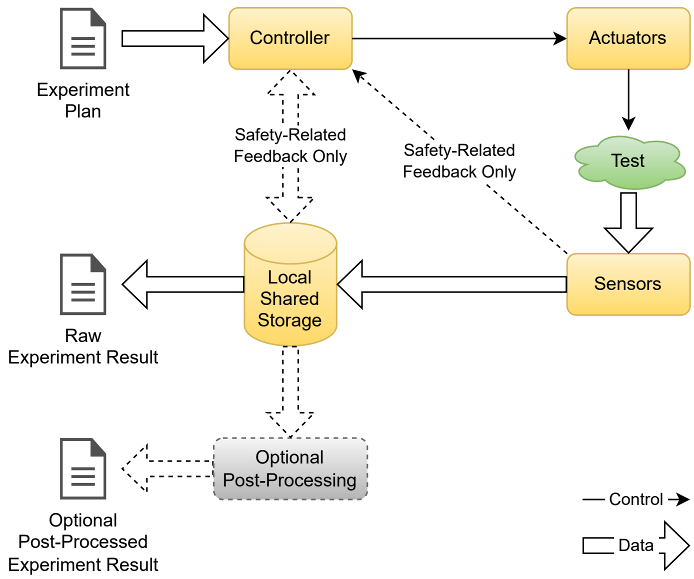


Figure 5-6. Local Experiment Control architectural pattern components and control/data flow

In contrast to this architectural pattern, the Distributed Experiment Control architectural pattern executes a predetermined plan using a remote experiment controller, i.e., with significant communication delay to remote components.

Examples: A robot-controlled chemistry laboratory science use case implements the Local Experiment Control strategic pattern, as a robot automates experiment execution. There is only safety-related feedback, such as to stop the robot in case of an emergency. The experiment controller, a workstation, is local. The individual pattern components are as follows:

- The experiment plan is the sequence of predetermined steps and associated parameters necessary to run the experiment. The predetermined steps include the parameters for synthesizing the chemical compound, route navigation instructions for the robot to move the sample between the different synthesis and characterization stations, parameters for characterizing the synthesized chemical compound, and safety related feedback instructions.
- The local experiment controller is a central workstation that is able to command and control the robot, synthesis equipment, analytical instruments, and any data and computing resources for analyzing the measurement data.
- The test performed in an experiment characterizes the synthesized chemical compound.
- The experiment result is a combination of the sample characterization results.

Known Uses: This strategic pattern is used in every experiment, where the predetermined experiment plan is automatically executed by a local controller. The only feedback is related to safety, such as to perform an emergency stop of the experiment when a chemistry laboratory is on fire. This pattern is the starting point for other patterns, as any feedback implemented by the Local Experiment Steering or Local Design of Experiment patterns and any orchestration of multiple experiments by the Local and Distributed Experiment Workflow patterns require the automatic execution of a predetermined experiment plan.

5.2.2 Distributed Experiment Control

Name: Distributed Experiment Control

Context: The pattern applies to a system with the following characteristics:

- An experiment plan exists that lists the predetermined actions to be performed while running the experiment.
- A remote experiment controller exists that executes the predetermined actions to be performed while running the experiment.
- Actuators may exist to allow for moving or controlling something before, during and/or after running the experiment.
- Sensors may exist to allow for measuring something before, during and/or after running the experiment.
- Instruments may exist that contain sensors and potentially actuators.
- Robots may exist that contain actuators and potentially sensors and that execute predetermined actions from the experiment plan in an automated or autonomous fashion.
- A component may exist that post-processes raw experiment data, such as to identify features.

Problem: Certain predetermined actions need to be performed while running an experiment. A remote experiment controller executes an experiment, incurring a potentially significant communication delay.

Forces: Only pre-experiment conditions are considered in performing the predetermined actions while running an experiment. Only safety-related conditions during the experiment may be considered. Other changing conditions during the experiment or post-experiment conditions are not considered.

Experiment actions are controlled with significant communication delay to remote components. Proper control capability does not need to be local, but must be able to act in time.

Solution: The is pattern implements the Experiment Control strategic pattern. A remote experiment controller executes an experiment using a predetermined experiment plan (Figure 5-7). The plan's execution is automated, performed in an open loop control and may involve human interaction. The controller may monitor the experiment for safety reasons. The plan contains a complete description of the predetermined actions to be performed for running the experiment, including any safety-related responses. Raw experiment data may be post-processed by an optional component, such as to identify features.

As the experiment controller is remote, component-local storage and explicit data transfer between components may be used for sensor and controller data. Control messages between these components orchestrate the control flow.

This pattern offers an open loop control with safety-related feedback on the experiment only. Experiment plan execution is automated, i.e., its list of actions is performed without external or human intervention that can unnecessarily hold up execution. Only 1 experiment is being controlled. There is a significant communication delay to remote components in the open loop control, as the experiment controller is remote.

Resulting Context: An experiment is executed automatically using a predetermined plan executed by a remote experiment controller, i.e., with significant communication delay to remote components.

Related Patterns: This architectural pattern implements the Experiment Control strategic pattern.

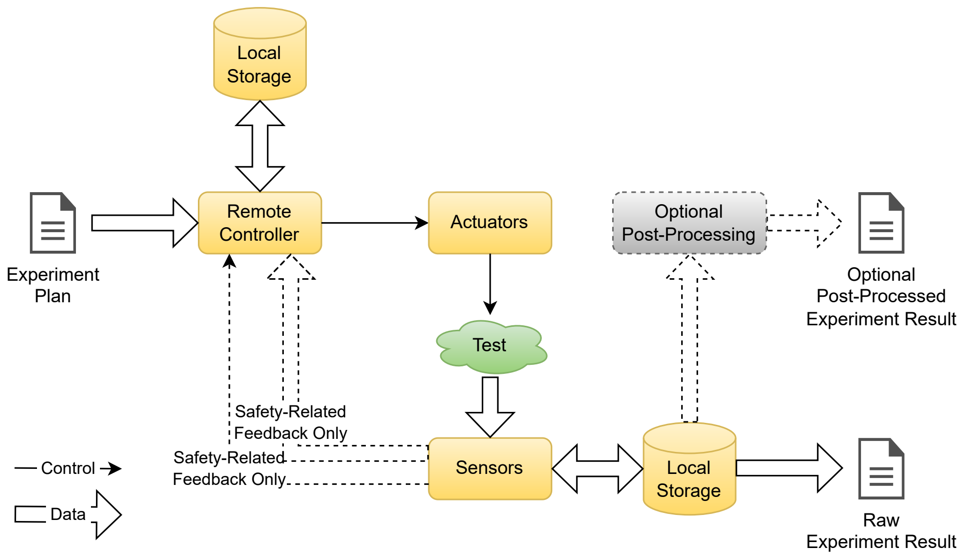


Figure 5-7. Distributed Experiment Control architectural pattern components and control/data flow

In contrast to this architectural pattern, the Local Experiment Control architectural pattern executes a predetermined plan using a local experiment controller, i.e., without significant communication delay to remote components.

Examples: A robot-controlled chemistry laboratory science use case can implement the Distributed Experiment Control strategic pattern, as a robot automates experiment execution. There is only safety-related feedback, such as to stop the robot in case of an emergency. The experiment controller, a scientist's laptop, is remote and connected to the robot-controlled chemistry laboratory over the Internet. The individual pattern components are as follows:

- The experiment plan is the sequence of predetermined steps and associated parameters necessary to run the experiment. The predetermined steps include the parameters for synthesizing the chemical compound, route navigation instructions for the robot to move the sample between the different synthesis and characterization stations, parameters for characterizing the synthesized chemical compound, and safety related feedback instructions.
- The remote experiment controller is a scientist's laptop that is able to command and control the robot, synthesis equipment, analytical instruments, and any data and computing resources for analyzing the measurement data.
- The test performed in an experiment characterizes the synthesized chemical compound.
- The experiment result is a combination of the sample characterization results.

Known Uses: This strategic pattern is used in every experiment, where the predetermined experiment plan is automatically executed by a remote controller. This is a rather rare use case, as the communication latency between a remote controller and the test is often too long. The only feedback is related to safety, such as to perform an emergency stop of the experiment when a chemistry laboratory is on fire. This pattern is the starting point for other patterns, as any feedback implemented by the Distributed Experiment Steering or Distributed Design of Experiment patterns and any orchestration of multiple experiments by the Local and Distributed Experiment Workflow patterns require the automatic execution of a predetermined experiment plan.

5.2.3 Local Experiment Steering

Name: Local Experiment Steering

Context: The pattern applies to a system with the following characteristics:

- An experiment plan exists that lists the predetermined actions to be performed while running the experiment, including potential parameter changes based on experiment progress.
- A local experiment controller exists that executes the predetermined actions to be performed while running the experiment.
- A local experiment analyzer exists that orients the observed information for the experiment controller.
- Sensors exist to allow for measuring experiment progress.
- Actuators may exist to allow for moving or controlling something before, during and/or after running the experiment.
- Additional sensors may exist to allow for measuring something before, during and/or after running the experiment.
- Instruments may exist that contain sensors and potentially actuators.
- Robots may exist that contain actuators and potentially sensors and that execute predetermined actions from the experiment plan in an automated or autonomous fashion.
- A component may exist that post-processes raw experiment data, such as to identify features.

Problem: Certain predetermined actions need to be performed while running an experiment to positively influence experiment progress. Experiment progress is analyzed and judged locally. There are no remote components that could incur a significant communication delay.

Forces: Only pre-experiment conditions and changing conditions during the experiment are considered in performing the predetermined actions while running an experiment. Post-experiment conditions are not considered.

Experiment progress is analyzed and judged without significant communication delay to remote components. Proper computational analysis and decision making capability must be present locally to be able to respond within a certain amount of time.

Solution: The is pattern implements the Experiment Steering strategic pattern using an OODA loop control. All components of the OODA loop control are local, i.e., physically located and connected in a way that does not incur a significant communication delay between the components.

As in the Experiment Steering strategic pattern, an experiment controller executes an experiment using a predetermined experiment plan and changes the plan's parameters during execution based on experiment progress (Figure 5-8). The plan's execution is autonomous, performed in a closed loop control and may involve human interaction. The controller may monitor the experiment for safety reasons. The plan contains a complete description of the predetermined actions to be performed for running the experiment, including any safety-related responses and how to analyze and judge experiment progress and change the plan accordingly. Raw experiment data may be post-processed by an optional component, such as to identify features.

The OODA loop control is formed by sensors that observe the experiment, an analyzer that orients the observed information, an experiment controller that decides on appropriate actions and actuators that perform the appropriate actions. As all components of the OODA loop control are local, a shared storage

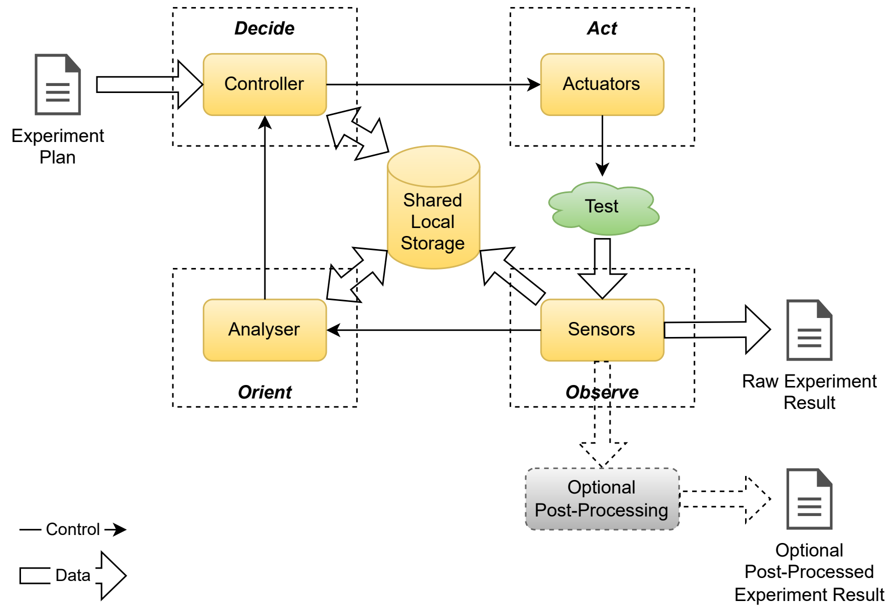


Figure 5-8. Local Experiment Steering architectural pattern components and control/data flow

device may be used between them for sensor, analyzer and controller data. Control messages between these components orchestrate the control flow.

This pattern offers a closed OODA loop control with safety-related feedback on the experiment and feedback on experiment progress. Experiment plan execution is autonomous, i.e., its list of actions changes during execution based on feedback and is performed without external or human intervention that can unnecessarily hold up execution. Only 1 experiment is being controlled. There is no significant communication delay to remote components in the closed OODA loop control, as the experiment progress analysis is local and the experiment controller is local as well.

Resulting Context: An experiment is executed autonomously using a predetermined experiment plan, with the plan's parameters changing autonomously during the experiment based on experiment progress. Experiment progress is analyzed and judged locally, i.e., without significant communication delay to remote components.

Related Patterns: This architectural pattern implements the Experiment Steering strategic pattern.

In contrast to this architectural pattern, the Distributed Experiment Steering architectural pattern analyzes and potentially also judges experiment progress remotely, i.e., with significant communication delay to remote components.

Examples: The Autonomous Continuous Flow Reactor Synthesis (AutoFlowS) science use case (Section 7.3) implements the Local Design of Experiments architectural pattern, as all components (planner, controller(s), synthesis station(s), and characterization station(s)) are local, i.e., in close physical and logical proximity with no significant latency (for communication or sample movement) to remote components. The experiment itself could be considered a Local Multi-Experiment Workflow architectural pattern using a sequence of Local Experiment Control architectural patterns. In this case, there is a significant overlap of the different components, as the same shared storage is being used, for example.

Known Uses: This architectural pattern is used in every experiment, where live feedback of locally analyzed experiment data is being used to autonomously change experiment parameters. A real-time feedback loop is feasible, as there are no remote components that could incur a significant communication delay.

5.2.4 Distributed Experiment Steering

Name: Distributed Experiment Steering

Context: The pattern applies to a system with the following characteristics:

- An experiment plan exists that lists the predetermined actions to be performed while running the experiment, including potential parameter changes based on experiment progress.
- A local or remote experiment controller exists that executes the predetermined actions to be performed while running the experiment.
- A remote experiment analyzer exists that orients the observed information for the experiment controller.
- Sensors exist to allow for measuring experiment progress.
- Actuators may exist to allow for moving or controlling something before, during and/or after running the experiment.
- Additional sensors may exist to allow for measuring something before, during and/or after running the experiment.
- Instruments may exist that contain sensors and potentially actuators.
- Robots may exist that contain actuators and potentially sensors and that execute predetermined actions from the experiment plan in an automated or autonomous fashion.
- A component may exist that post-processes raw experiment data, such as to identify features.

Problem: Certain predetermined actions need to be performed while running an experiment to positively influence experiment progress. Experiment progress is analyzed and optionally also judged/controlled remotely, incurring a potentially significant communication delay.

Forces: Only pre-experiment conditions and changing conditions during the experiment are considered in performing the predetermined actions while running an experiment. Post-experiment conditions are not considered.

Experiment progress is analyzed and optionally also judged with significant communication delay to remote components. Proper computational analysis and decision making capability does not need to be local, but must be able to respond within a certain amount of time.

Solution: The is pattern implements the Experiment Steering strategic pattern using an OODA loop control. The *Orient* component and optionally the *Decide* component of the of the OODA loop control are remote, i.e., physically located and connected in a way that does incur a significant communication delay between the components.

As in the Experiment Steering strategic pattern, an experiment controller executes an experiment using a predetermined experiment plan and changes the plan's parameters during execution based on experiment progress (Figure 5-9). The plan's execution is autonomous, performed in a closed loop control and may involve human interaction. The controller may monitor the experiment for safety reasons. The plan contains a complete description of the predetermined action to be performed for running the experiment, including any safety-related responses and how to analyze and judge experiment progress and change the plan accordingly. Raw experiment data may be post-processed by an optional component, such as to identify features.

The OODA loop control is formed by sensors that observe the experiment, an analyzer that orients the observed information, an experiment controller that decides on appropriate actions and actuators that

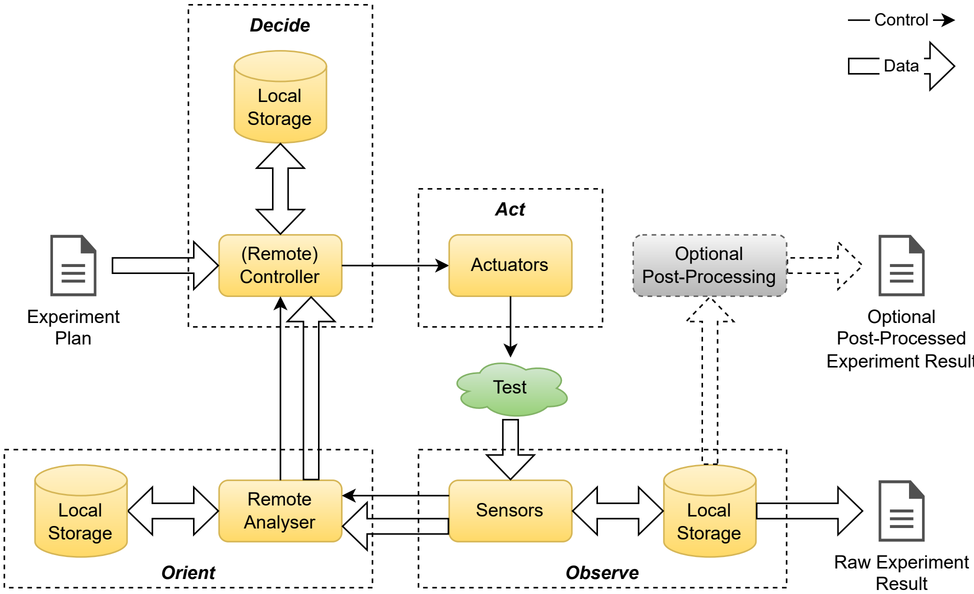


Figure 5-9. Distributed Experiment Steering architectural pattern components and control/data flow

perform the appropriate actions. As some components of the OODA loop control are remote, component-local storage and explicit data transfer between components may be used for sensor, analyzer and controller data. Control messages between these components orchestrate the control flow.

This pattern offers a closed OODA loop control with safety-related feedback on the experiment and feedback on experiment progress. Experiment plan execution is autonomous, i.e., its list of actions changes during execution based on feedback and is performed without external or human intervention that can unnecessarily hold up execution. Only 1 experiment is being controlled. There is a significant communication delay to remote components in the closed OODA loop control, as the experiment progress analysis is remotely and the experiment controller may be remote as well.

Resulting Context: An experiment is executed autonomously using a predetermined experiment plan, with the plan's parameters changing autonomously during the experiment based on experiment progress. Experiment progress is analyzed and potentially also judged remotely, i.e., with significant communication delay to remote components.

Related Patterns: This architectural pattern implements the Experiment Steering strategic pattern.

In contrast to this architectural pattern, the Local Experiment Steering architectural pattern analyzes and judges experiment progress locally, i.e., without significant communication delay to remote components.

Examples: An autonomous microscopy science use case [16] (Section 7.4) implements the Distributed Experiment Steering architectural pattern, as an ongoing STEM experiment is controlled by remote analyses of periodic experiment data. At the architectural pattern level of abstraction, the individual pattern components are as follows:

- In addition to the properties identified by the Experiment Steering strategic pattern, the experiment controller is either local or remote and may feature a graphical user interface (GUI) or some other human-machine interface (HMI).

- The actuator is part of the STEM and moves the scanning electron beam.
- The test is performed in the STEM experiment determines the properties of microscopic structures.
- The sensor is part of the STEM and provides the raw microscope data.

Known Uses: This architectural pattern is used in every experiment, where live feedback of remotely analyzed experiment data is being used to autonomously change experiment parameters. Given the potentially significant communication delay between the experiment and the remote analysis, real-time feedback loop solutions are limited.

5.2.5 Local Design of Experiments

Name: Local Design of Experiments

Context: The pattern applies to a system with the following characteristics:

- An experiment design plan exists that lists the predetermined actions to be performed for creating a new experiment plan based on prior experiment results.
- An initial experiment plan exists that lists the predetermined actions to be performed while running the experiment.
- A local experiment planner exists that creates the new experiment plan based on prior experiment results.
- A local experiment controller exists that executes the predetermined actions to be performed while running the experiment.
- A local experiment analyzer exists that orients the observed information for the experiment planner.
- Sensors exist to allow for measuring experiment results.
- Actuators may exist to allow for moving or controlling something before, during and/or after running the experiment.
- Additional sensors may exist to allow for measuring something before, during and/or after running the experiment.
- Instruments may exist that contain sensors and potentially actuators.
- Robots may exist that contain actuators and potentially sensors and that execute predetermined actions from the experiment plan in an automated or autonomous fashion.
- A component may exist that post-processes raw experiment data, such as to identify features.

Problem: Certain predetermined actions need to be performed to run a set of similar experiments with different experiment plan parameters, depending on experiment results. Experiment results are analyzed and judged locally. There are no remote components that could incur a significant communication delay.

Forces: Only pre- and post-experiment conditions are considered in performing the predetermined actions to run a set of similar experiments with different experiment plan parameters. Only safety-related conditions during the experiment may be considered. Other changing conditions during the experiments are not considered, unless the Experiment Steering strategic pattern is being used in conjunction with this architectural pattern, such as by using the Local Experiment Steering or Distributed Experiment Steering architectural patterns.

Experiment results are analyzed and judged without significant communication delay to remote components. Proper computational analysis and decision making capability must be present locally to be able to respond within a certain amount of time.

Solution: The is pattern implements the Design of Experiments strategic pattern using an OODA loop control. All components of the OODA loop control are local, i.e., physically located and connected in a way that does not incur a significant communication delay between the components.

As in the Design of Experiments strategic pattern, an experiment controller executes each experiment using a predetermined experiment plan (Figure 5-10). The plan's execution is automated, performed in an open loop control and may involve human interaction. The controller may monitor the experiment for safety reasons. The experiment plan contains a complete description of the predetermined actions to be performed for running the experiment, including any safety-related responses. An experiment planner creates the experiment plan, based on an experiment design plan and prior experiment results (if any). The experiment plan change is autonomous, performed in a closed loop control and may involve human interaction. The experiment design plan contains an initial experiment plan and a plan for creating new experiment plans based on experiment results, including how to analyze and judge experiment results and change the plan accordingly. Raw experiment data may be post-processed by an optional component, such as to identify features.

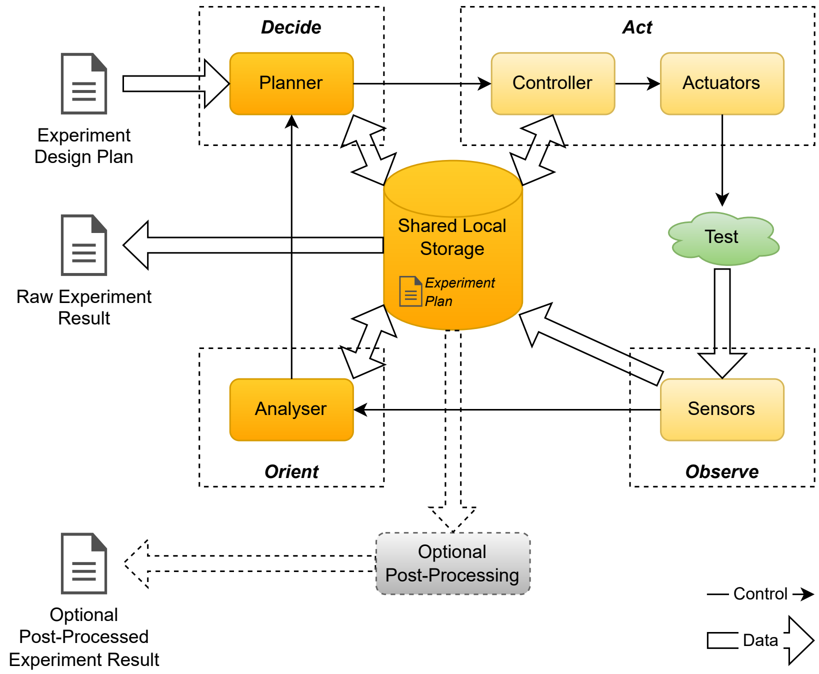


Figure 5-10. Local Design of Experiments architectural pattern components and control/data flow

The OODA loop control is formed by sensors that observe the experiment, an analyzer that orients the observed information, an experiment planner that decides on appropriate actions, and an experiment controller and actuators that perform the appropriate actions. As all components of the OODA loop control are local, a shared storage device may be used between them for sensor, analyzer, planner and controller data. Control messages between these components orchestrate the control flow.

This pattern offers an open loop control with safety-related feedback on the experiment and a separate closed OODA loop control with feedback on experiment results. Experiment plan execution is automated within the open loop control, i.e., its list of actions is performed without external or human intervention that can unnecessarily hold up execution. Experiment design plan execution is autonomous, i.e., it creates a new experiment plan after each experiment based on experiment results and is performed without external or human intervention that can unnecessarily hold up execution. A set of similar experiments with different experiment plan parameters is controlled. There is no significant communication delay to remote components in the open loop control, as the experiment controller is local. There is also no significant communication delay to remote components in the closed OODA loop control, as the experiment result analysis and experiment planner are local as well.

Resulting Context: An experiment is executed autonomously with different experiment plan parameters using a predetermined experiment plan, with the plan's parameters changing autonomously between experiments based on experiment results. Experiment results are analyzed and judged locally, i.e., without significant communication delay to remote components.

Related Patterns: This architectural pattern implements the Design of Experiments strategic pattern. It relies on the Experiment Control strategic pattern for automatically executing a predetermined experiment plan. This architectural pattern can be extended using the Experiment Steering strategic pattern (instead of the Experiment Control strategic pattern) for autonomously executing a predetermined experiment plan, with the plan's parameters changing autonomously during experiments based on experiment progress. Such extension may involve the Local Experiment Steering or Distributed Experiment Steering architectural patterns.

In contrast to this architectural pattern, the Distributed Design of Experiments architectural pattern analyzes and potentially also judges experiment results remotely, i.e., with significant communication delay to remote components.

Examples: An ACL science use case (Section 7.5) implements the Local Design of Experiments architectural pattern, as all components (planner, controller(s), robot, synthesis station(s), and characterization station(s)) are local, i.e., in close physical and logical proximity with no significant latency (for communication or sample movement) to remote components.

The experiment is a complex sequence of steps involving multiple instruments, actuators, sensors, etc. Thus, the experiment itself could be considered a Local Multi-Experiment Workflow architectural pattern using a sequence of Local Experiment Control architectural patterns. Examples of steps that constitute the Multi-Experiment Workflow architecture include the synthesis step and each of the individual characterization steps, such as the gas chromatography, high performance liquid chromatography, and X-ray microscopy. Some of these steps could potentially be performed in parallel if the sample were broken down into pieces such that the pieces could be analyzed by the characterization instruments in parallel. There is a significant overlap of the different components of the patterns, as the same shared storage is being used, for example.

Known Uses: This architectural pattern is used in every experiment, where feedback of experiment results is being used to autonomously change the parameters of the next experiment(s) using components that are all local, i.e., in close physical and logical proximity with no significant latency (for communication or sample movement) to remote components. Known uses range from having simple linear or random parameter scan to complex probabilistic approaches (e.g., Bayesian design of experiments) or domain science informed AI (e.g., physics-informed design of experiments) in the feedback loop.

5.2.6 Distributed Design of Experiments

Name: Distributed Design of Experiments

Context: The pattern applies to a system with the following characteristics:

- An experiment design plan exists that lists the predetermined actions to be performed for creating a new experiment plan based on prior experiment results.
- An initial experiment plan exists that lists the predetermined actions to be performed while running the experiment.
- A local or remote experiment planner exists that creates the new experiment plan based on prior experiment results.
- A local experiment controller exists that executes the predetermined actions to be performed while running the experiment.
- A remote experiment analyzer exists that orients the observed information for the experiment planner.
- Sensors exist to allow for measuring experiment results.
- Actuators may exist to allow for moving or controlling something before, during and/or after running the experiment.
- Additional sensors may exist to allow for measuring something before, during and/or after running the experiment.
- Instruments may exist that contain sensors and potentially actuators.
- Robots may exist that contain actuators and potentially sensors and that execute predetermined actions from the experiment plan in an automated or autonomous fashion.
- A component may exist that post-processes raw experiment data, such as to identify features.

Problem: Certain predetermined actions need to be performed to run a set of similar experiments with different experiment plan parameters, depending on experiment results. Experiment results are analyzed and optionally also judged/controlled remotely, incurring a potentially significant communication delay.

Forces: Only pre- and post-experiment conditions are considered in performing the predetermined actions to run a set of similar experiments with different experiment plan parameters. Only safety-related conditions during the experiment may be considered. Other changing conditions during the experiments are not considered, unless the Experiment Steering strategic pattern is being used in conjunction with this architectural pattern, such as by using the Local Experiment Steering or Distributed Experiment Steering architectural patterns.

Experiment results are analyzed and optionally also judged with significant communication delay to remote components. Proper computational analysis and decision making capability does not need to be local, but must be able to respond within a certain amount of time.

Solution: The is pattern implements the Design of Experiments strategic pattern using an OODA loop control. The *Orient* component and optionally the *Decide* component of the of the OODA loop control are remote, i.e., physically located and connected in a way that does incur a significant communication delay between the components.

As in the Design of Experiments strategic pattern, an experiment controller executes each experiment using a predetermined experiment plan (Figure 5-11). The plan's execution is automated, performed in an open loop control and may involve human interaction. The controller may monitor the experiment for safety reasons. The experiment plan contains a complete description of the predetermined actions to be performed for running the experiment, including any safety-related responses. An experiment planner creates the experiment plan, based on an experiment design plan and prior experiment results (if any). The experiment plan change is autonomous, performed in a closed loop control and may involve human interaction. The experiment design plan contains an initial experiment plan and a plan for creating new experiment plans based on experiment results, including how to analyze and judge experiment results and change the plan accordingly. Raw experiment data may be post-processed by an optional component, such as to identify features.

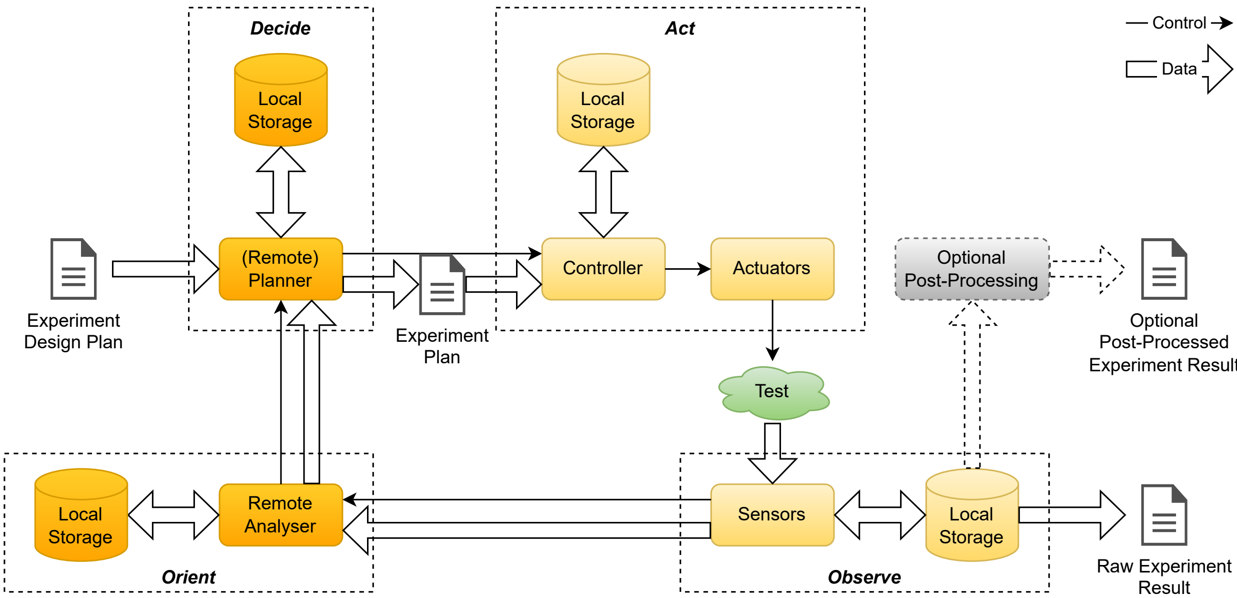


Figure 5-11. Distributed Design of Experiments architectural pattern components and control/data flow

The OODA loop control is formed by sensors that observe the experiment, an analyzer that orients the observed information, an experiment planner that decides on appropriate actions, and an experiment controller and actuators that perform the appropriate actions. As some components of the OODA loop control are remote, component-local storage and explicit data transfer between components may be used for sensor, analyzer, planner and controller data. Control messages between these components orchestrate the control flow.

This pattern offers an open loop control with safety-related feedback on the experiment and a separate closed OODA loop control with feedback on experiment results. Experiment plan execution is automated within the open loop control, i.e., its list of actions is performed without external or human intervention that can unnecessarily hold up execution. Experiment design plan execution is autonomous, i.e., it creates a new experiment plan after each experiment based on experiment results and is performed without external or human intervention that can unnecessarily hold up execution. A set of similar experiments with different experiment plan parameters is controlled. There is no significant communication delay to remote components in the open loop control, as the experiment controller is local. There is a significant communication delay to remote components in the closed OODA loop control, as the experiment results are analyzed remotely and the experiment planner may be remote as well.

Resulting Context: An experiment is executed autonomously with different experiment plan parameters using a predetermined experiment plan, with the plan's parameters changing autonomously between experiments based on experiment results. Experiment results are analyzed and potentially also judged remotely, i.e., with significant communication delay to remote components.

Related Patterns: This architectural pattern implements the Design of Experiments strategic pattern. It relies on the Experiment Control strategic pattern for automatically executing a predetermined experiment plan. This architectural pattern can be extended using the Experiment Steering strategic pattern (instead of the Experiment Control strategic pattern) for autonomously executing a predetermined experiment plan, with the plan's parameters changing autonomously during experiments based on experiment progress. Such extension may involve the Local Experiment Steering or Distributed Experiment Steering architectural patterns.

In contrast to this architectural pattern, the Local Design of Experiments architectural pattern analyzes and judges experiment results locally, i.e., without significant communication delay to remote components.

Examples: An ACL science use case (Section 7.5) implements the Local Design of Experiments architectural pattern, but could implement the Distributed Design of Experiments architectural pattern if additional analysis is performed remotely, such as using an AI-based physics-informed digital twin of the experiment.

The experiment is a complex sequence of steps involving multiple instruments, actuators, sensors, etc. Thus, the experiment itself could be considered a Local Multi-Experiment Workflow architectural pattern using a sequence of Local Experiment Control architectural patterns. Examples of steps that constitute the Multi-Experiment Workflow architecture include the synthesis step and each of the individual characterization steps, such as the gas chromatography, high performance liquid chromatography, and X-ray microscopy. Some of these steps could potentially be performed in parallel if the sample were broken down into pieces such that the pieces could be analyzed by the characterization instruments in parallel. There is a significant overlap of the different components of the patterns, as the same shared storage is being used, for example.

Known Uses: This architectural pattern is used in every experiment, where feedback of experiment results is being used to autonomously change the parameters of the next experiment(s) using components that are remote, i.e., not in close physical and logical proximity and with some significant latency (for communication or sample movement) to remote components. Known uses range from having simple linear or random parameter scan to complex probabilistic approaches (e.g., Bayesian design of experiments) or domain science informed AI (e.g., physics-informed design of experiments) in the feedback loop.

5.2.7 Local Multi-Experiment Workflow

Name: Local Multi-Experiment Workflow

Context: The pattern applies to a system with the following characteristics:

- A multi-experiment workflow plan exists that lists the predetermined actions to be performed for executing each experiment plan.
- An experiment plan exists for each experiment that lists the predetermined actions to be performed while running the experiment.
- Local actuators may exist to allow for moving or controlling something before, during and/or after running the experiment.
- Local sensors may exist to allow for measuring something before, during and/or after running the experiment.
- Local instruments may exist that contain sensors and potentially actuators.
- Local robots may exist that contain actuators and potentially sensors and that execute predetermined actions from the experiment plan in an automated or autonomous fashion.

Problem: Certain predetermined actions need to be performed to run a set of experiments in serial (one after another) and/or in parallel (simultaneously). All experiments, i.e., their tests, are local . There are no remote experiments, i.e., their tests that could incur a significant communication delay between dependent experiments or to the multi-experiment workflow controller. Local experiments may involve remote components, such as by using the Distributed Experiment Control, Distributed Experiment Steering or Distributed Design of Experiments architectural patterns.

Forces: Only pre- and post-experiment conditions are considered in performing the predetermined actions to run the set of experiments in serial and/or parallel. Safety-related conditions during the experiments may be considered. Only pre-experiment conditions are considered in performing the predetermined actions while running each experiment individually, unless the Experiment Steering strategic pattern or the Design of Experiments strategic pattern are being used for some or all experiments. If the Experiment Steering strategic pattern is being used for a particular experiment, then changing conditions during this experiment are considered in performing the predetermined actions while running it. If the Design of Experiments strategic pattern is being used for a particular experiment, then post-experiment conditions are considered in performing the predetermined actions to run it with different experiment plan parameters.

Individual experiment execution is not impacted by a significant communication delay, as all experiments are local. Multi-experiment workflow scheduling, execution and completion is not impacted by a significant communication delay between remote experiments and the multi-experiment workflow controller.

Solution: A multi-experiment workflow controller orchestrates the execution of the experiments using a predetermined multi-experiment workflow plan (Figure 5-12). The multi-experiment workflow plan's execution is automated, performed in an open loop control and may involve human interaction. The multi-experiment workflow controller may monitor one or more experiment controllers for dependency reasons. The multi-experiment workflow plan contains a complete description of the predetermined actions to be performed for orchestrating the execution of the experiments including any dependency-related responses.

Multiple local experiment controllers execute their experiments using their predetermined experiment plan.

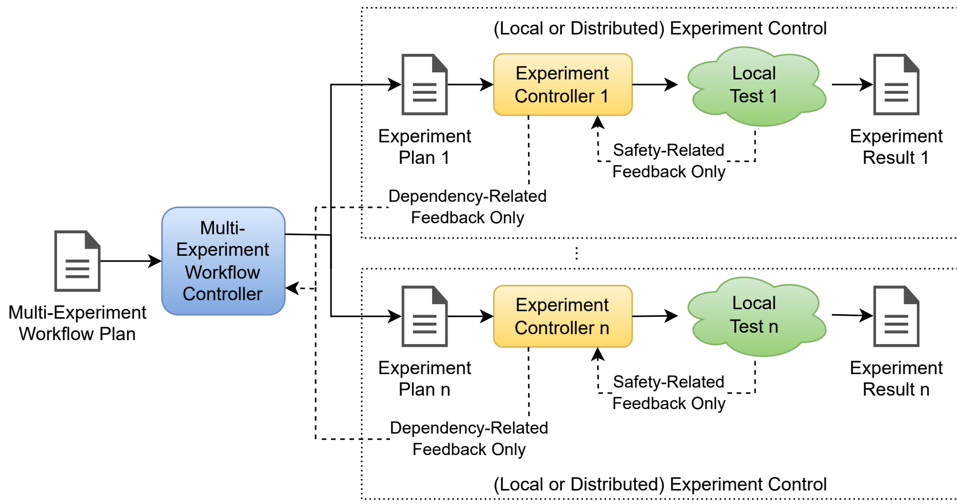


Figure 5-12. Local Multi-Experiment Workflow architectural pattern components and control/data flow (using multiple local experiments with Local or Distributed Experiment Control architectural patterns in this example)

Each plan's execution is automated, performed in an open loop control and may involve human interaction. Each experiment controller may monitor the experiment for safety reasons. Each experiment plan contains a complete description of the predetermined actions to be performed for running its experiment, including any safety-related responses.

Some experiments may be executed in parallel, as they do not depend on each other, while other experiment may be executed in serial due to dependencies. The orchestration of the execution follows a DAG with the experiments as vertices and the edges as dependencies (Figure 5-13). A dependency between experiments may arise when one experiment needs the result of another. All experiments are local in the DAG.

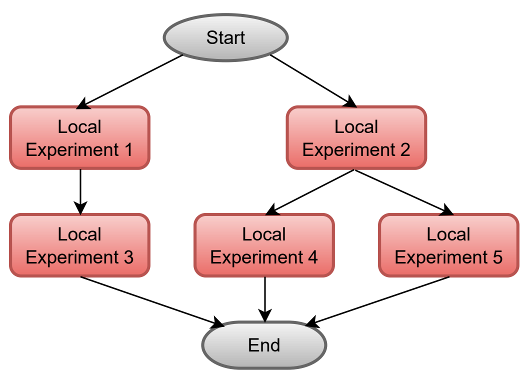


Figure 5-13. Example of a Local Multi-Experiment Workflow architectural pattern directed acyclic graph

This pattern offers an open loop control with safety-related feedback on each experiment and a separate loop control with safety-related feedback for each experiment. Experiment plan execution is automated within the open loop control for each experiment, i.e., its list of actions is performed without external or human intervention that can unnecessarily hold up execution. Multi-experiment workflow plan execution is automated within the open loop control for all experiments, i.e., its list of actions is performed without external or human intervention that can unnecessarily hold up execution. A set of serial and/or parallel experiments is controlled. All experiments are local and there is no significant communication delay between dependent experiments or to the multi-experiment workflow controller.

Resulting Context: Experiments are executed automatically in serial and/or parallel using a predetermined plan locally, i.e., without significant communication delay between dependent experiments or to the multi-experiment workflow controller.

Related Patterns: This strategic pattern relies on the Experiment Control strategic pattern for automatically executing each predetermined experiment plan. This strategic pattern can be extended using the Experiment Steering strategic pattern (instead of the Experiment Control strategic pattern) for autonomously executing some or all predetermined experiment plans, with each plan's parameters changing autonomously during experiments based on progress. This strategic pattern can also be extended using the Design of Experiments strategic pattern for autonomously executing some or all predetermined experiment plans, with each plan's parameters changing autonomously between experiments based on results. The Experiment Control, Experiment Steering and Design of Experiments strategic patterns can be used together in conjunction with this strategic pattern, individually for each experiment of the multi-experiment workflow. However, the Experiment Control and Experiment Steering strategic patterns are mutually exclusive for the same experiment, as the Experiment Steering strategic pattern extends the Experiment Control strategic pattern.

In contrast to this pattern, the Distributed Multi-Experiment Workflow architectural pattern executes experiments that are local and remote, i.e., with significant communication delay between dependent experiments or to the multi-experiment workflow controller.

Examples: In the ACL science use case (Section 7.5), the experiment itself could be considered a Local Multi-Experiment Workflow architectural pattern using a sequence of Local Experiment Control architectural patterns. In this case, there is a significant overlap of the different components, as the same shared storage is being used, for example.

Known Uses: This strategic pattern is used every time a set of experiments are performed that are local to each other. Very common examples are (1) a set of parallel local experiments that investigate the same physical sample that is getting split up beforehand, or (2) a set of serial local experiments that investigate the same physical sample that is getting moved from one experiment to the next with no delay. Each of these experiments investigates different properties, where the overall combination of the experiment results may be part of a bigger experiment that encompasses them.

5.2.8 Distributed Multi-Experiment Workflow

Name: Distributed Multi-Experiment Workflow

Context: The pattern applies to a system with the following characteristics:

- A multi-experiment workflow plan exists that lists the predetermined actions to be performed for executing each experiment plan.
- An experiment plan exists for each experiment that lists the predetermined actions to be performed while running the experiment
- Local and remote actuators may exist to allow for moving or controlling something before, during and/or after running the experiment.
- Local and remote sensors may exist to allow for measuring something before, during and/or after running the experiment.
- Local and remote instruments may exist that contain sensors and potentially actuators.
- Local and remote robots may exist that contain actuators and potentially sensors and that execute predetermined actions from the experiment plan in an automated or autonomous fashion.

Problem: Certain predetermined actions need to be performed to run a set of experiments in serial (one after another) and/or in parallel (simultaneously). One or more experiments, i.e., their tests, are remote, incurring a potentially significant communication delay between dependent experiments or to the multi-experiment workflow controller. Local experiments may involve remote components, such as by using the Distributed Experiment Control, Distributed Experiment Steering or Distributed Design of Experiments architectural patterns. Remote experiments may involve components that are completely local to the remote location, such as by using the Local Experiment Control, Local Experiment Steering or Local Design of Experiments architectural patterns.

Forces: Only pre- and post-experiment conditions are considered in performing the predetermined actions to run the set of experiments in serial and/or parallel. Safety-related conditions during the experiments may be considered. Only pre-experiment conditions are considered in performing the predetermined actions while running each experiment individually, unless the Experiment Steering strategic pattern or the Design of Experiments strategic pattern are being used for some or all experiments. If the Experiment Steering strategic pattern is being used for a particular experiment, then changing conditions during this experiment are considered in performing the predetermined actions while running it. If the Design of Experiments strategic pattern is being used for a particular experiment, then post-experiment conditions are considered in performing the predetermined actions to run it with different experiment plan parameters.

Individual experiment execution may be impacted by a significant communication delay to a set of remote experiments it depends on. Multi-experiment workflow scheduling, execution and completion may be impacted by a significant communication delay between remote experiments and the multi-experiment workflow controller.

Solution: A multi-experiment workflow controller orchestrates the execution of the experiments using a predetermined multi-experiment workflow plan (Figure 5-14). The multi-experiment workflow plan's execution is automated, performed in an open loop control and may involve human interaction. The multi-experiment workflow controller may monitor one or more experiment controllers for dependency reasons. The multi-experiment workflow plan contains a complete description of the predetermined actions to be performed for orchestrating the execution of the experiments including any dependency-related responses.

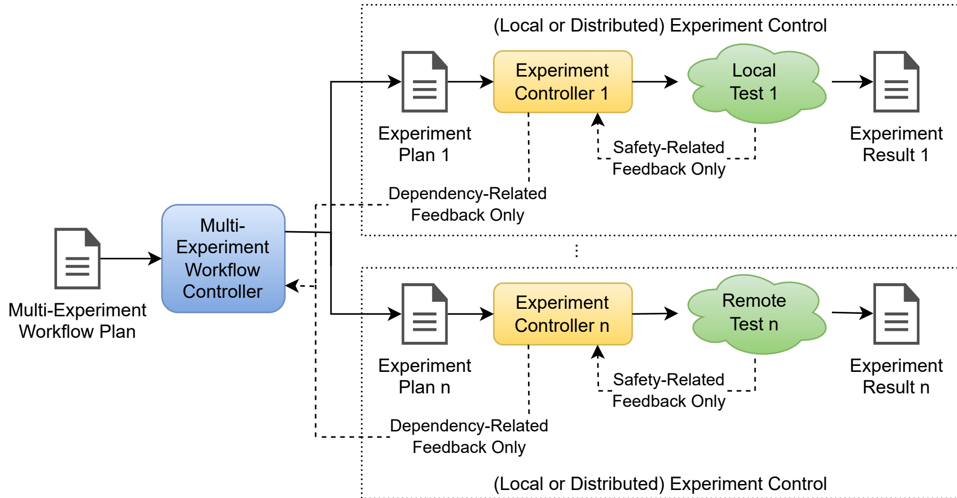


Figure 5-14. Distributed Multi-Experiment Workflow architectural pattern components and control/data flow (using multiple local and remote experiments with Local or Distributed Experiment Control architectural patterns in this example)

Multiple local and/or remote experiment controllers execute their experiments using their predetermined experiment plan. Each plan's execution is automated, performed in an open loop control and may involve human interaction. Each experiment controller may monitor the experiment for safety reasons. Each experiment plan contains a complete description of the predetermined actions to be performed for running its experiment, including any safety-related responses.

Some experiments may be executed in parallel, as they do not depend on each other, while other experiment may be executed in serial due to dependencies. The orchestration of the execution follows a DAG with the experiments as vertices and the edges as dependencies (Figure 5-15). A dependency between experiments may arise when one experiment needs the result of another. Some experiments are remote in the DAG.

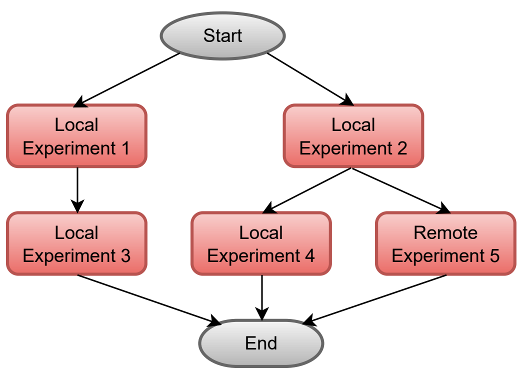


Figure 5-15. Example of a Distributed Multi-Experiment Workflow architectural pattern directed acyclic graph

This pattern offers an open loop control with safety-related feedback on each experiment and a separate loop control with safety-related feedback for each experiment. Experiment plan execution is automated within the open loop control for each experiment, i.e., its list of actions is performed without external or human intervention that can unnecessarily hold up execution. Multi-experiment workflow plan execution is automated within the open loop control for all experiments, i.e., its list of actions is performed without external or human intervention that can unnecessarily hold up execution. A set of serial and/or parallel experiments is controlled. Some experiments are remote and there is a significant communication delay between dependent experiments or to the multi-experiment workflow controller.

Resulting Context: Experiments are executed automatically in serial and/or parallel using a predetermined plan locally and remotely, i.e., with significant communication delay between dependent experiments or to the multi-experiment workflow controller.

Related Patterns: This strategic pattern relies on the Experiment Control strategic pattern for automatically executing each predetermined experiment plan. This strategic pattern can be extended using the Experiment Steering strategic pattern (instead of the Experiment Control strategic pattern) for autonomously executing some or all predetermined experiment plans, with each plan's parameters changing autonomously during experiments based on progress. This strategic pattern can also be extended using the Design of Experiments strategic pattern for autonomously executing some or all predetermined experiment plans, with each plan's parameters changing autonomously between experiments based on results. The Experiment Control, Experiment Steering and Design of Experiments strategic patterns can be used together in conjunction with this strategic pattern, individually for each experiment of the multi-experiment workflow. However, the Experiment Control and Experiment Steering strategic patterns are mutually exclusive for the same experiment, as the Experiment Steering strategic pattern extends the Experiment Control strategic pattern.

In contrast to this pattern, the Local Multi-Experiment Workflow architectural pattern executes experiments that are local only, i.e., without significant communication delay between dependent experiments or to the multi-experiment workflow controller.

Examples: A science use case involving an ACL and an automated neutron diffraction experiment that further characterizes a compound, such as a grown crystal, implements a Distributed Multi-Experiment Workflow architectural pattern. The tests of both experiments are remote to each other. However, they do depend on each other as the ACL produces and initially characterizes a compound and the Spallation Neutron Source (SNS) further characterizes it. The two experiments are executed serially, requiring transportation of the compound from the first to the second.

Known Uses: This strategic pattern is used every time a set of experiments are performed, where one or more of them are considered remote. Very common examples are (1) a set of parallel experiments that investigate the same physical sample that is getting split up beforehand, or (2) a set of serial experiments that investigate the same physical sample that is getting moved from one experiment to the next with some delay due to physical distances. Each of these experiments investigates different properties, where the overall combination of the experiment results may be part of a bigger experiment that encompasses them.

6. BUILDING SOLUTIONS USING SCIENCE USE CASE DESIGN PATTERNS

The science use case design patterns detailed in the catalog focus on the inherent open or closed loop control problem as a common problem to be solved in reducing human-in-the-loop requirements with machine-in-the-loop capabilities. Scientific instruments, robot-controlled laboratories and edge/center computing/data resources are connected in a loop control to enable autonomous experiments, self-driving laboratories, smart manufacturing, and AI-driven design, discovery and evaluation. Each science use case design pattern consists of a behavior and a set of interfaces in the context of performing a single or a set of experiments in an open or closed loop control. The abstract design pattern definitions describe solutions free of implementation details.

The science use case design patterns are divided into two different classes: (1) strategic patterns that define high-level solution methods using experiment control architecture features at a very coarse granularity, and (2) architectural patterns that define more specific solution methods using hardware and software architecture features at a finer granularity. The architectural patterns inherit the features of their parent strategic patterns, but also address additional problems that are not exposed at the high abstraction level of the strategic patterns.

6.1 A STEP-BY-STEP GUIDE

Building a complete solution from an existing science use case requires dissecting the science use case by the open or closed loop control problem or problems it contains. This Section 6.1 describes the involved steps and discusses the individual decision parameters in more detail. The following Section 6.2 discusses additional considerations when composing different design patterns, such as due to multiple loop control problems. Each loop control problem needs to be identified, including its properties and hardware/software architectural features. A step-by-step decomposition process would work as follows:

1. Clearly define the experiment or experiments that are being performed
2. Identify the loop control problem or problems that exist for each experiment.
3. Classify each loop control problem by a strategic pattern.
4. Identify the individual components of each loop control problem and associated strategic pattern.
5. Classify each loop control problem by an architectural pattern that matches its strategic pattern.
6. Match the identified components with the components of the architectural patterns.
7. Design the hardware/software architecture of the solution based on the architectural patterns and the corresponding matched components, using the pattern properties as design guidelines.

What is the experiment?: It is important to clearly define the experiment or experiments, as the wrong definition ultimately leads the designer down the wrong path. It is often easier to think of an experiment as a concrete test process that demonstrates a specific known truth, examines the validity of a specific hypothesis, or determines specific properties of something. Clearly identifying the experiment devices, such as sensors, actuators, instruments and robots, is part of that definition as well. It is quite possible that one experiment in a laboratory tries to accomplish multiple objectives, in which case a single multi-objective experiment could be split up into multiple experiments, especially if it involves a workflow or completely separated loop control problems. There is no hard rule on this and any such split would be on a case-by-case basis.

Which loop control problems exist?: Separating out what is being controlled and how is the key to identifying the loop control problem or problems that exist for each experiment. In pretty much all cases, there is some type of simple open loop control, as described in the Experiment Control strategic pattern. Additional loop control problems may exist that may extend the simple open loop control, such as to the Experiment Steering strategic pattern, or uses/relies on the simple open loop control, such as with the Design of Experiments strategic pattern. There also may be multiple loop control problems for the same experiment, such as a combination of the the Experiment Steering and Design of Experiments strategic patterns. Similarly, a multi-objective experiment may have multiple loop control problems for different parts of the experiment, potentially requiring it to be split up into multiple experiment. Obviously, a multi-experiment workflow may have loop control problems for each experiment in the workflow. Pattern combinations that solve such issues are discussed in Section 6.2.

Who is in control?: The science use case design patterns have one controller component and some have an additional planner component. These are not necessarily physical standalone components. Instead, an analyzer may already contain the decision-making logic and also act as a controller or planner. Similarly, the controller or planner may require human input or may be a human itself. While the goal is to reduce human-in-the-loop requirements with machine-in-the-loop capabilities, this may be a process that requires a transition and some human-in-the-loop requirements may not necessarily completely eliminated.

Which strategic pattern?: The key differences in features between the 4 strategic patterns are (1) no feedback, (2) feedback for the same experiment, (3) feedback for the next experiment, and (4) workflow of multiple experiments. If there is no feedback, then Experiment Control is the right strategic pattern. If there is feedback for the same experiment, such as changing a parameter based on a measurement to observe how that or another measurement changes, then Experiment Steering is the right strategic pattern. If there is feedback for the next experiment, such as to change the parameters and re-run the experiment, then Design of Experiments is the right strategic pattern. There are experiments, where the experiment plan constantly evolves as the experiment is performed, based on measurements. In this case, either Experiment Steering or Design of Experiments may be used, whichever is closer. In this case, using Design of Experiments splits the experiment into multiple separate experiments with different experiment plans. Multi-Experiment Workflow is used whenever there are multiple experiments without feedback. There could be a greater feedback loop over multiple experiments in a workflow. In this case, a separate strategic pattern is employed (see Section 6.2).

What is local? What is remote?: The architectural science use case design patterns distinguish between local and remote components based on communication delay. Any potentially significant communication delay to a component makes it a remote component. The term "significant communication delay" is purposely not clearly defined to give designers room for interpretation. There may be other reasons for defining a component as remote, such as when a component is physically located at an entirely different location that does not necessarily incur a significant communication delay but requires a special way of communication. A human that acts as a planner and communicates with the rest of the system via e-mail or a GUI would likely also be considered a remote component.

6.2 PATTERN COMPOSITIONS

A solution may require the composition of science use case design patterns. A simple example from the pattern catalog is the Design of Experiments strategic pattern that already uses the Experiment Control strategic pattern, but could use the Experiment Steering strategic pattern instead. Similarly, the Multi-Experiment Workflow strategic pattern already uses the Experiment Control strategic pattern, but could use the Experiment Steering strategic pattern, the Design of Experiments strategic patterns, or a combination of Experiment Control, Experiment Steering and Design of Experiments strategic patterns instead. This composition of strategic patterns is then also reflected in composition of architectural patterns.

The decision to compose a solution from multiple science use case design patterns depends on the actual properties of the solution. The most significant indicator is the need for multiple, different control loops. Another indicator is the existence of a Multi-Experiment Workflow with different experiments that have different control loops. The number and properties of the control loops typically define the composition of science use case design patterns, from strategy to architectural. Note that there may be more than one control loop implementing the same strategic and even architectural pattern, but with different properties. For example, there may be multiple Local Experiment Steering control loops that are independent from each other. They may operate with different timing requirements, perform analysis on different computational resources and modify different parameters independent from each other.

The following example illustrates the composition of science use case design patterns. In this solution, there is a control loop for Experiment Steering to change parameters based on observation as the experiment is progressing. There is also a second control loop for Design of Experiments to change the Experiment Plan based on the prior experiment result after each experiment. Figure 6-1 illustrates the involved components and control/data flow of the Experiment Steering and the Design of Experiments strategic pattern composition. The Experiment Design Plan and the Experiment Planner are exclusive parts of the Design of Experiments strategic pattern, while the other components are part of the Experiment Steering strategic pattern that the Design of Experiments strategic pattern is using as its experiment to control from an Experiment Plan perspective.

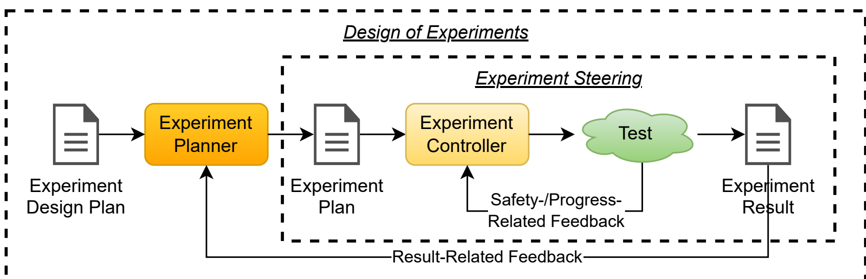


Figure 6-1. Example: Components and control/data flow of Experiment Steering and Design of Experiments strategic pattern composition

In the given science use case example, the Experiment Steering utilizes a local shared storage device, such as a small network attached storage (NAS), for all sensor data and its analysis results. It also relies on a local computational resource, such as an NVIDIA Jetson Nano, for analysis and decision making. The Design of Experiments transfers the sensor data of the entire experiment from the shared storage device to a remote analyzer, such as an NVIDIA DGX system. Its analysis results are evaluated and a new experiment plan is created by the Controller on a desktop system running a GUI. The corresponding involved components and control/data flow of the Local Experiment Steering and the Distributed Design of Experiments architectural pattern composition is shown in Figure 6-2.

This is just an example of how a solution may require the composition of science use case design patterns. Different logical components may utilize the same physical components, such as when different control loops use the same storage device or the same computational resource for analysis and/or control. For example, separate controllers for different Experiment Steering control loops may use exactly the same

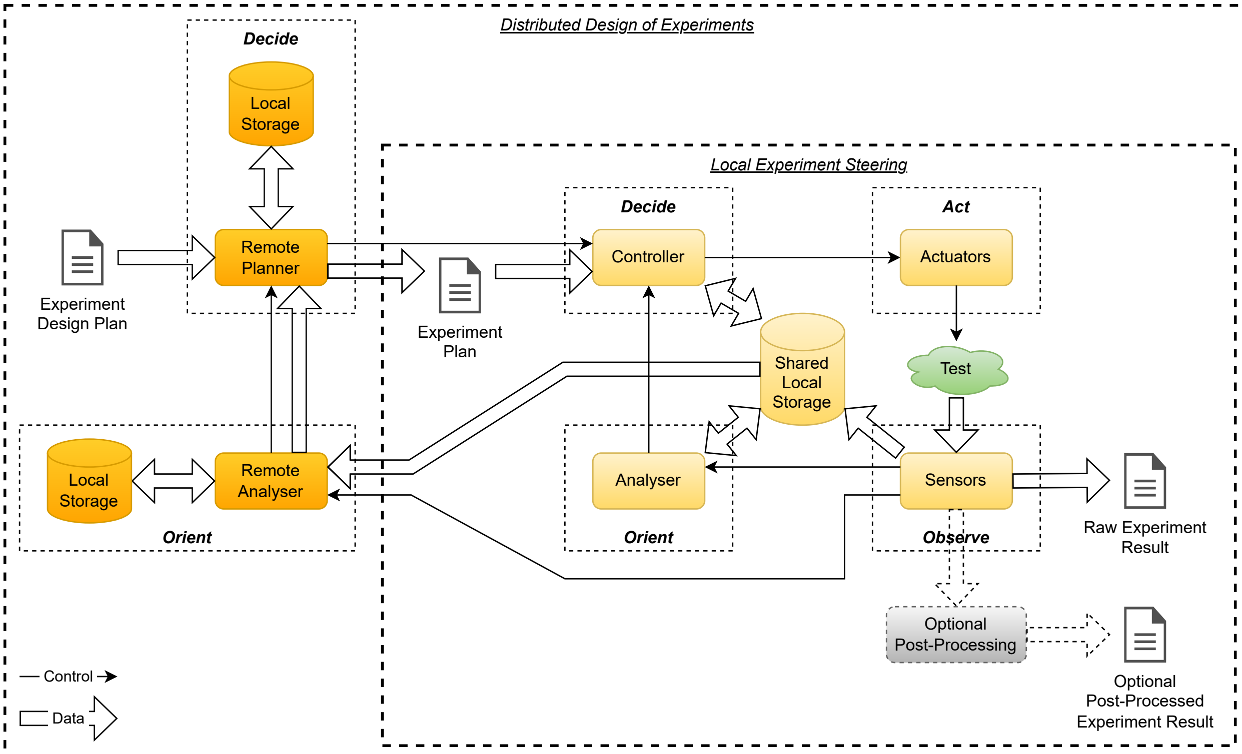


Figure 6-2. Example: Components and control/data flow of Local Experiment Steering and Distributed Design of Experiments architectural pattern composition

physical component, such as a Raspberry Pi, for storing and analyzing sensor data and for issuing different, non-conflicting control commands to a robot.

7. CASE STUDIES

7.1 AUTOMATION FOR GRID INTERCONNECTED-LABORATORY EMULATION

The Automation for Grid Interconnected-Laboratory Emulation (AGILE) project (Figure 7-1) addresses the challenge of emulating and automating the emulation of the real-world energy system and power grid at the Grid Research Integration and Deployment Center (GRID-C) laboratories. The emulation includes scalable power grid dynamic simulation in real-time edge computing resources, a small portion of the power grid represented in laboratories through digital twinning resources (and control nodes), and secure stable low latency data exchange between these resources. The automation includes mapping of real-world energy system and power grid to scalable power grid dynamic simulation and partitioning in different GRID-C laboratories.

In the recent years, equipment, such as energy storage system (ESS), photovoltaic (PV) generation systems, inverters, fast electric vehicle (EV) chargers, electrical loads, transformers, among others, have been introduced in different GRID-C laboratories. Individual components, such as ESS or PV systems or EV chargers have been researched upon as a stand-alone system, without considering the interactions between multiple next-generation technology and/or power grid interactions. The latter is important to de-risk transfer of knowledge and technology to field. Developing the emulation of real-world energy system and power grid through this project will advance the capabilities at GRID-C and will be an enabler for it to become a one-of-its-kind self-driven automated smart laboratory in future.

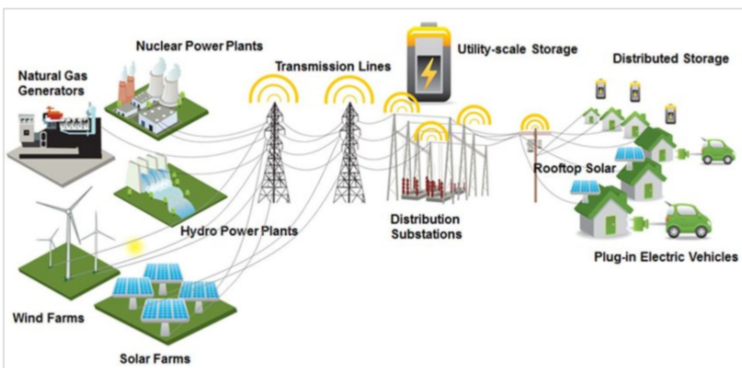


Figure 7-1. The Automation for Grid Interconnected-Laboratory Emulation use case

The AGILE science use case implements the Experiment Steering strategic pattern (Figure 7-2), as the experiment is the emulation of a real-world energy system and power grid that uses additional simulation at different granularities in a real-time feedback loop for steering the emulation. At the strategic pattern level of abstraction, the individual pattern components are as follows:

- The experiment plan describes the overarching interaction of power electronics (PE) nodes in the GRID-C laboratory performing the emulation of the US power grid.
- The experiment controller is supervising and regulating the GRID-C PE nodes.
- The test performed in an experiment characterizes the emulated US power grid.
- The experiment result is the power and current data gathered from the GRID-C PE nodes.

The AGILE science use case implements the Local Experiment Steering architectural pattern (Figure 7-3), as an ongoing emulation of a real-world energy system and power grid is guided by a local analysis of

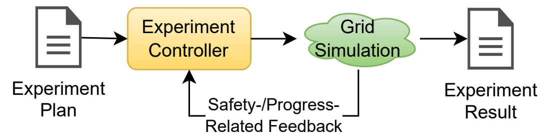


Figure 7-2. Experiment Steering strategic pattern for the AGILE science use case

frequent periodic real-time experiment data in a simulation. At the architectural pattern level of abstraction, the individual pattern components are as follows:

- In addition to the properties identified by the Experiment Steering strategic pattern, the local experiment controller supervises and regulates the GRID-C PE nodes in real time.
- The local analysis component is a separate NVIDIA DGX system that runs the additional emulation/simulation at different granularities with real-time feedback to the controller.

Although different networked systems are used or control, analysis, and emulation, this science use case follows the Local Experiment Steering architectural pattern, as the control and data flow has real-time characteristics in the microsecond range.

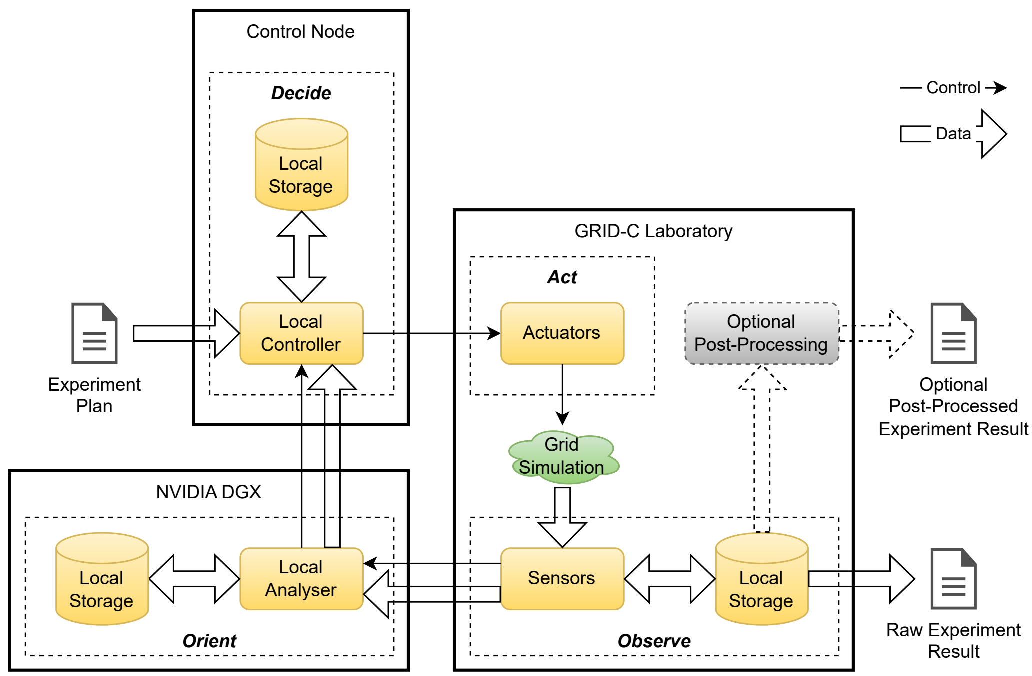


Figure 7-3. Local Experiment Steering architectural pattern for the AGILE science use case

7.2 AUTONOMOUS ADDITIVE MANUFACTURING

Automation and autonomy can enable revolutionary scientific advances by coordinating a diverse array of experimental and computational capabilities more efficiently and more effectively than current hands-on approaches. This project creates an autonomous system to plan and adaptively control additive manufacturing (AM) build processes (Figure 7-4). It involves multiple characterization modes, computation across the edge-to-center computing continuum, and multiple scientific user facilities. The objective of the autonomous AM system is to control the residual stress in a part to address a grand challenge — building parts that are ready and safe to use immediately (i.e., "born qualified"). This project enables secure, automated, time-sensitive interactions between experimental and computational components. It demonstrates a new method for autonomous control, combining in-situ observations and thermo-mechanical simulations for accurate real-time state estimation. It uses thermo-mechanical simulations in the control loop to predict the complex, long-range effects of 3D metal printing process parameters on part quality. This autonomous system enables AM builds with residual stress at least two times closer to the desired distribution than current methods, drastically reducing the time to develop process parameters for new alloys and geometries.

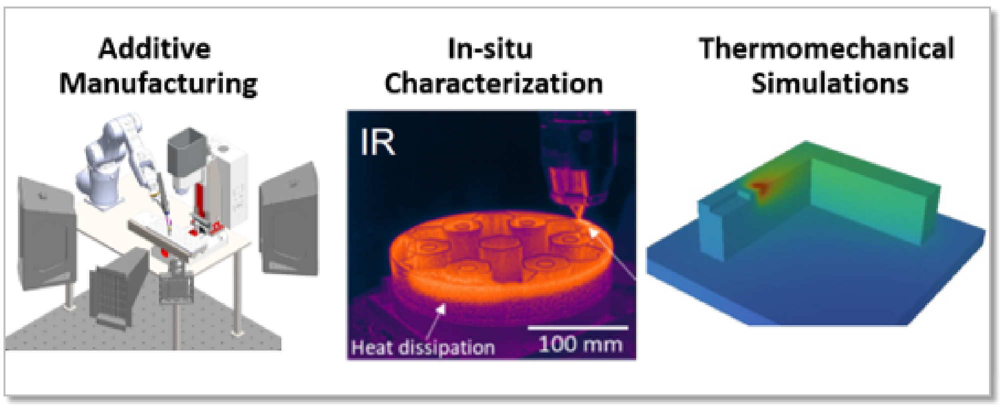


Figure 7-4. The INTERSECT additive manufacturing system performs 3D metal printing with in-situ observations and thermo-mechanical simulations to build "born qualified" structures.

The INTERSECT additive manufacturing system has several loop control problems (Figure 7-5). The first loop control implements an Experiment Steering strategic pattern. It obtains temperature data from thermocouple sensors mounted to the base of the printed object and from an infra-red (IR) camera that is observing the printing process from an angle above. The temperature data is streamed to an analyzer that performs faster-than-realtime structural simulation of the stresses created by the repeated heating and cooling cycles of the printed material. The structural simulation data is used to change the parameters of the laser on the next printed layer. This permits adapting the live printing process to the simulated stresses inferred by the measured data. The second loop control involves the neutron beam of Oak Ridge National Laboratory (ORNL)'s Spallation Neutron Source (SNS) to obtain more detailed and multi-scale structural data. The neutron diffraction measurements and corresponding digital image correlation of the entire 3D printing process permit validation of the structural simulation and adaptation of the 3D printing process for the next part to be printed in an Experiment Design strategic pattern. At the strategic pattern level of abstraction, the individual pattern components are as follows:

- The experiment design plan describes the goal, which is the validated 3D printing of a metal part with predetermined structural stresses.

- The experiment planner is the subject matter expert (SME) that validates the structural simulation using neutron diffraction data and correspondingly adjusts the experiment components and plan.
- The experiment plan is the sequence of predetermined steps and associated parameters necessary to 3D print the metal part. The parameters include the targeted structural stress and the options for changing the laser parameters, such as temperature and speed.
- The experiment controller is the control computer system of the 3D metal printer.
- The test performed in an experiment 3D prints a metal part that conforms to given structural stress parameters.
- The experiment result consists of (1) the 3D printed metal part, (1) the corresponding thermal data, (3) the structural simulation data inferring the stresses in the part, and (4) the raw and analyzed neutron diffraction data for validation.

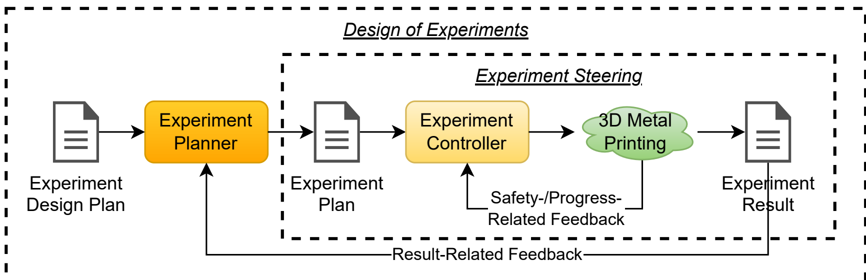


Figure 7-5. Experiment Steering and Design of Experiments strategic patterns for the AM science use case

The The INTERSECT additive manufacturing system (Figure 7-6) implements the Distributed Experiment Steering architectural pattern, as the analyzer that performs the structural simulation of the stresses is a remote NVIDIA DGX computing system, the Summit supercomputer, or the Frontier supercomputer, depending on simulation accuracy and speed needs and corresponding computational requirements. It further implements the Distributed Design of Experiments architectural pattern, as the analyzer that performs the image correlation and validation of the structural simulation is a remote computing system as well. It may even involve two different remote computing systems, one for the image correlation and one for the validation.

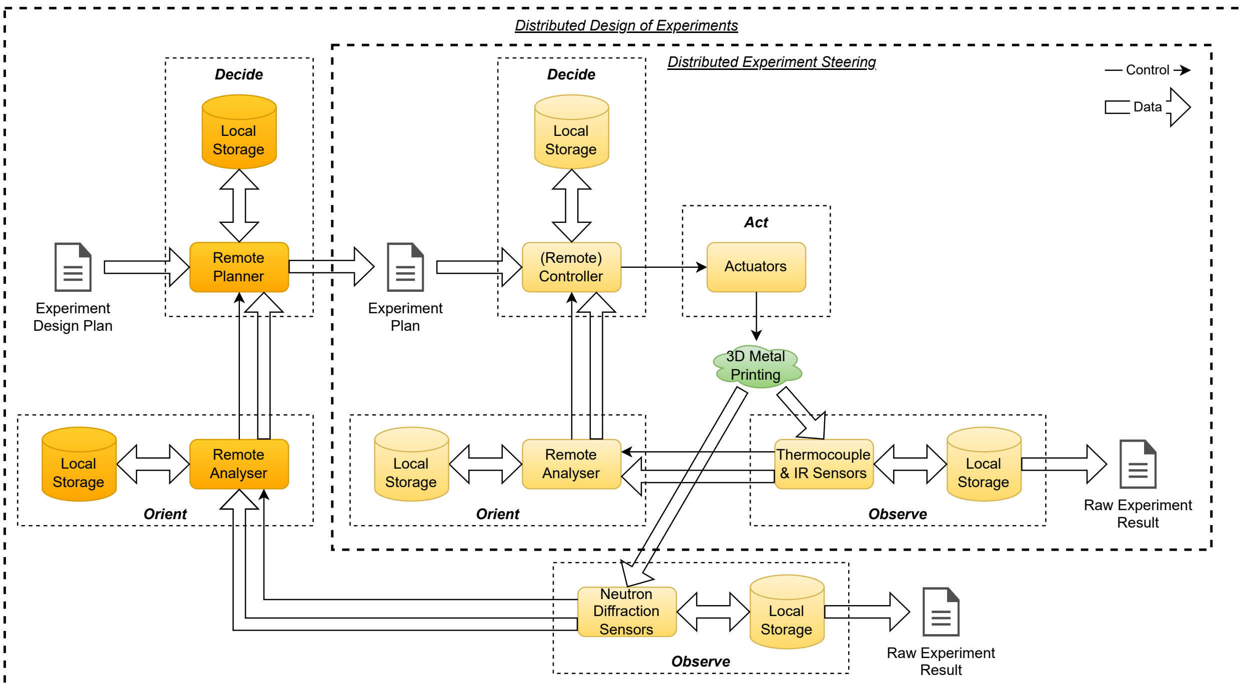


Figure 7-6. Distributed Experiment Steering and Distributed Design of Experiments strategic patterns for the AM science use case

7.3 AUTONOMOUS CONTINUOUS FLOW REACTOR SYNTHESIS

There is a critical need for efficient scale-up of atom precise synthesis products to quantities that enable full characterization of the structure-composition-property relationships and for scalable deployment of important material for key applications. Coupled with new instrumentation design, robotics, and in-operando interconnected analytical tools, automation, intelligent discovery of synthesis pathways is feasible and can potentially bridge the gap for scale-up of new materials. Autonomous Continuous Flow Reactor Synthesis (AutoFlowS) (Figure 7-7) is a system targeting this capability using an autonomous continuous flow chemistry framework that translates high-quality lead molecules and materials to quantities that meet scalability demands. At its core the continuous flow synthesis platform can design its own viable synthesis pathway to a particular molecule or material and then autonomously carry it out. Ultimately the goal is to enable enhanced automation of all aspects of the scientific discovery process: from hypotheses generation to the design of experiments to testing hypotheses to execution of physical experiments to the analysis and interpretation of the results.

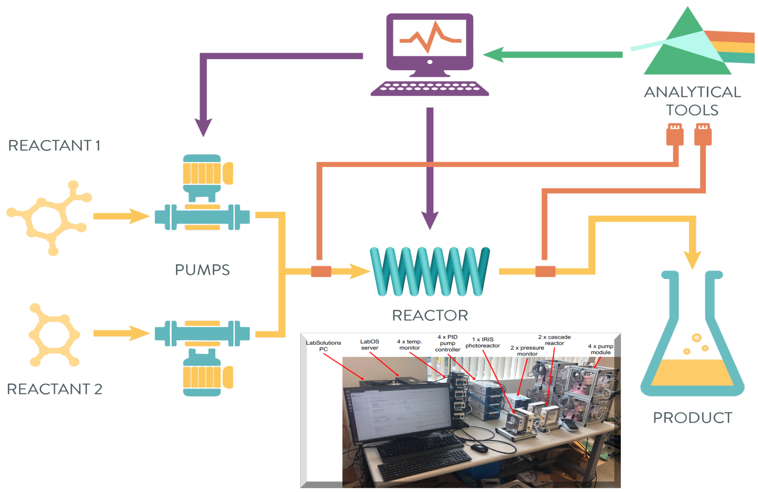


Figure 7-7. An autonomous continuous flow reactor synthesis science use case

The AutoFlowS science use case implements the Design of Experiments strategic pattern (Figure 7-8), as a continuous flow reactor performs experiments, experiment results are analyzed, and subsequent experiments are performed based on analysis results. At the strategic pattern level of abstraction, the individual pattern components are as follows:

- The experiment design plan describes the goal, which is the desired chemical compound, and the logic necessary to craft subsequent experiments towards achieving the goal.
- The experiment planner could be the SME who could be substituted by a machine learning or deep learning model implemented in the SnapDragon control software which would decide on the next experiment plan given the experiment results from past experiments.
- The experiment plan would be the sequence of predetermined steps and associated parameters necessary to run the current experiment. The predetermined steps include the parameters for the programmable valves to release the correct amounts of the desired reactant chemicals from the vials, parameters to control the reaction in the reactor, parameters for characterizing the synthesized chemical compound, and safety related feedback instructions.
- The SnapDragon software is the experiment controller, communicating with and controlling the

pumps, reactor, and analytical tools.

- The test performed in an experiment characterizes the synthesized chemical compound.
- The experiment result is a combination of the sample characterization results.

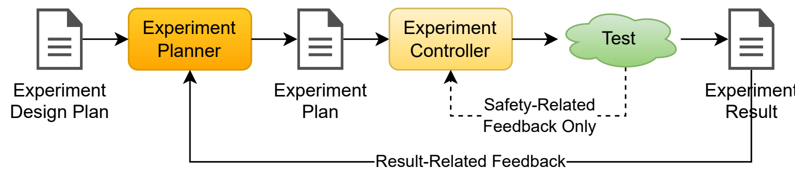


Figure 7-8. Design of Experiments strategic pattern for the AutoFlowS science use case

The experiment is a complex sequence of steps involving multiple instruments i.e., reactant vials and tools), actuators (i.e., pumps and valves), sensors (i.e., analytical tools), etc. Thus, the Experiment itself could be considered a Multi-Experiment Workflow strategic pattern or a sequence of Experiment Control strategic patterns (Figure 7-9). Examples of steps that constitute the Multi-Experiment Workflow strategic pattern include the control of the pumps for reactants, controlling the reactor, and each of the individual characterization steps such as the infrared spectroscopy, Raman spectroscopy, non-magnetic resonance imaging, fluorescence imaging, quartz crystal micro-balance measurements, viscosity meters, etc. Many of these steps could potentially be performed in parallel since the chemical product can be channeled to different analytical tools simultaneously.

The AutoFlowS science use case implements the Local Design of Experiments architectural pattern (Figure 7-10), as all components (planner, controller(s), synthesis station(s), and characterization station(s)) are local, i.e., in close physical and logical proximity with no significant latency (for communication or sample movement) to remote components. The experiment itself could be considered a Local Multi-Experiment Workflow architectural pattern using a sequence of Local Experiment Control architectural patterns (Figure 7-11). In this case, there is a significant overlap of the different components, as the same shared storage is being used, for example.

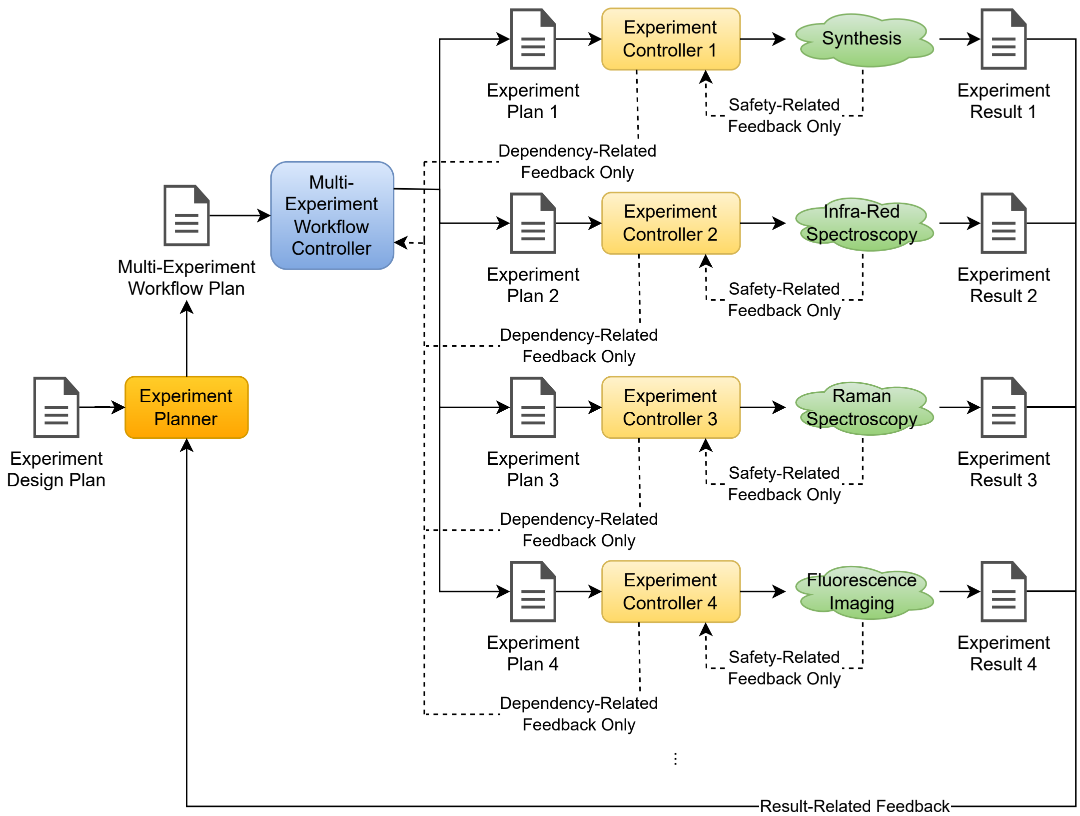


Figure 7-9. Design of Experiments strategic pattern for the AutoFlowS science use case, using the Multi-Experiment Workflow strategic pattern.

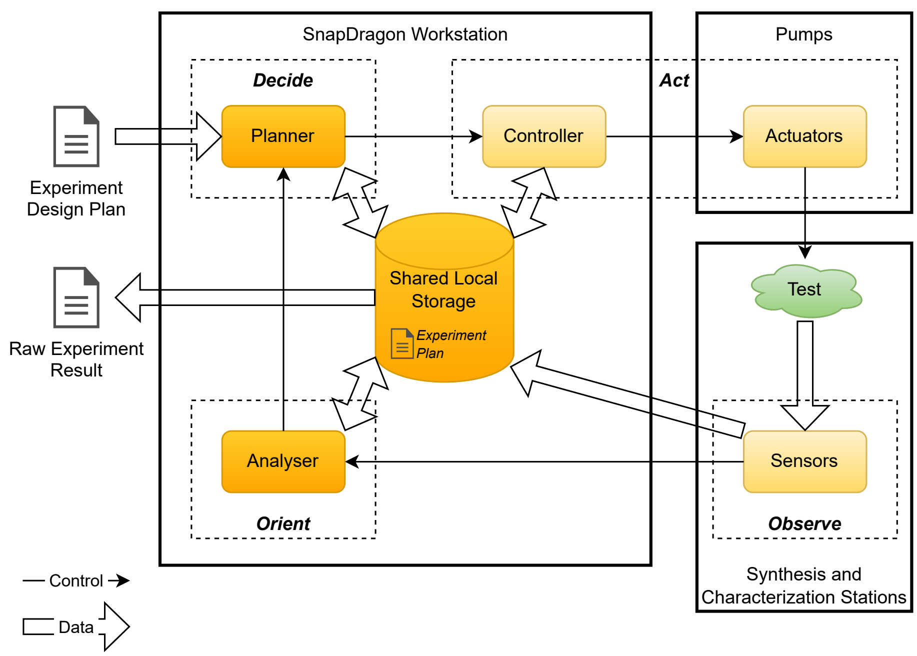


Figure 7-10. Local Design of Experiments architectural pattern for the AutoFlowS science use case

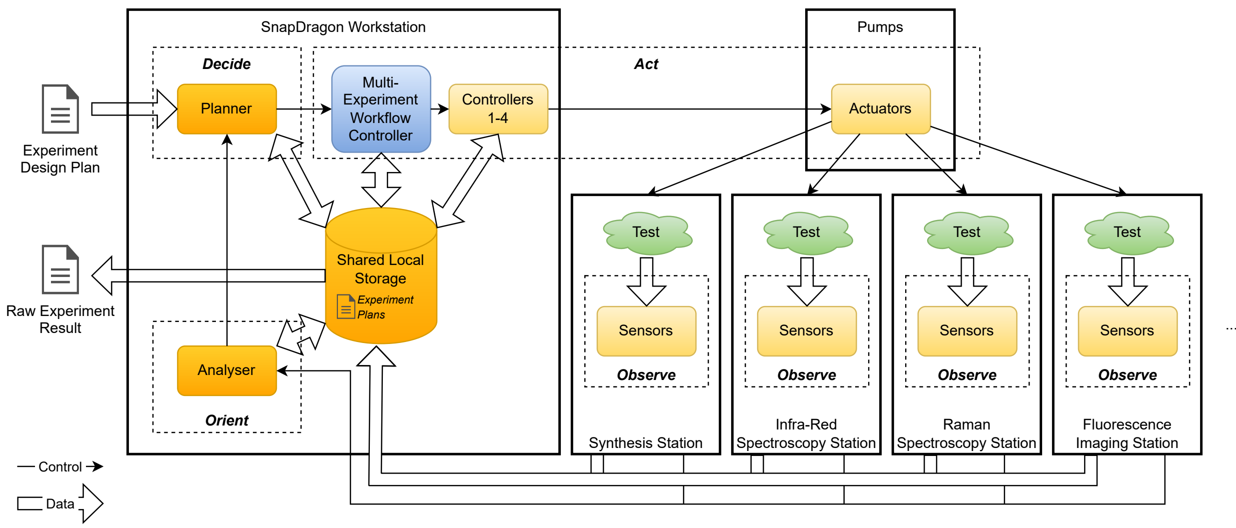


Figure 7-11. Local Design of Experiments architectural pattern for the AutoFlowS science use case, using the Local Multi-Experiment Workflow architectural pattern

7.4 AUTONOMOUS MICROSCOPY

Over the last two decades, aberration corrected scanning transmission electron microscopy (STEM) has become the mainstay of condensed matter physics, materials science, chemistry, catalysis, and nanotechnology. Yet both deriving fundamental physical insights from these multidimensional data sets and further using this knowledge towards making new and better materials for energy, quantum, and information technologies is stymied by the dearth of the methods and the data infrastructure necessary to handle the bespoke data sets. Similarly, human-driven operation of the microscopes is inherently limited at the age of fast data acquisition and multidimensional data. This project establishes the crucial link between the cutting-edge instrumental platforms at the Center for Nanophase Materials Sciences (CNMS) and the high-performance computing (HPC) capabilities at ORNL to enable the next generation of the HPC-driven scientific discovery (Figure 7-12). Specifically, the project targets three intertwined tasks, including (a) establishing a pipeline for direct data transfer from the STEM to the molecular dynamics (MD)/density functional theory (DFT) environment, (b) high-fidelity structural reconstructions from 4D STEM data, and (c) enabling the science-driven physical discovery via deep kernel learning. These targets, while closely linked scientifically, offer complementary challenges for the data infrastructure, prioritizing central processing unit (CPU) based calculations, general-purpose computing graphics processing unit (GPGPU) and high-volume data transfer, and low-latency CPU operations respectively.

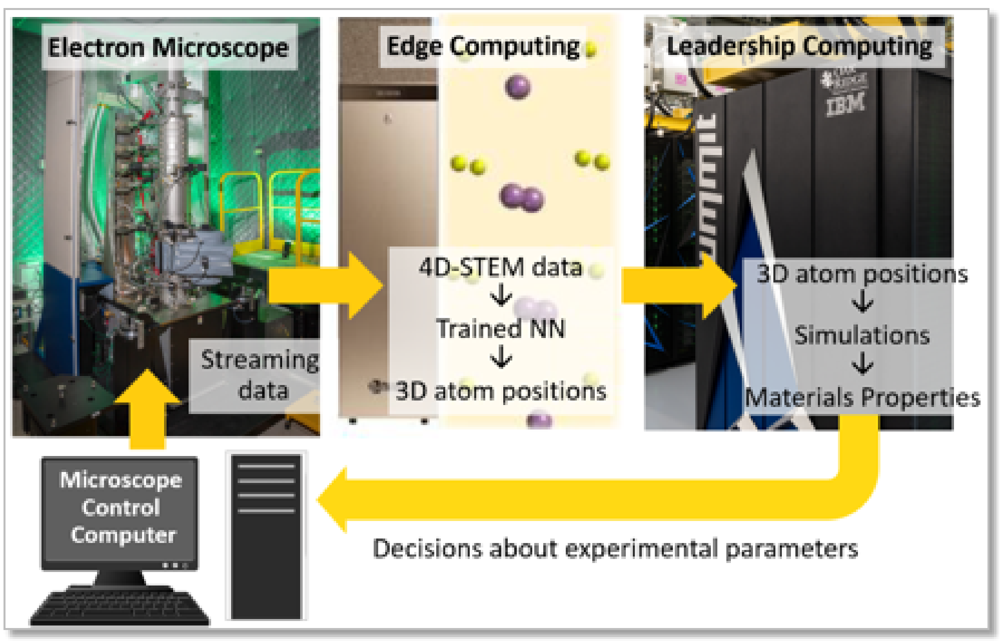


Figure 7-12. An autonomous microscopy science use case

The autonomous microscopy science use case [16] implements the Experiment Steering strategic pattern (Figure 7-13), as an ongoing STEM experiment is controlled by analyses of periodic experiment data. At the strategic pattern level of abstraction, the individual pattern components are as follows:

- The experiment plan contains a complete description of the predetermined actions to be performed for running the experiment, including any parameters for operating the STEM, safety-related responses and how to analyze and judge experiment progress and change the plan accordingly. The experiment plan also contains the goal of the STEM experiment to steer it in the right direction and to stop its closed loop control upon completion.

- The experiment controller executes an experiment using a predetermined experiment plan and changes the plan's parameters during execution based on experiment progress. The plan's execution is autonomous, performed in a closed loop control and may involve human interaction.
- The test performed in an experiment determines the properties of microscopic structures.
- The experiment result is a combination of raw and analyzed STEM data and insights derived from this data.

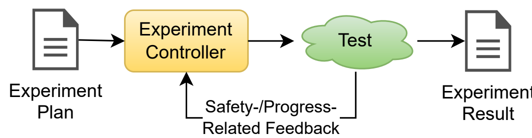


Figure 7-13. Experiment Steering strategic pattern for the autonomous microscopy science use case

The autonomous microscopy science use case implements the Distributed Experiment Steering architectural pattern (Figure 7-14), as an ongoing STEM experiment is controlled by remote analyses of periodic experiment data. At the architectural pattern level of abstraction, the individual pattern components are as follows:

- In addition to the properties identified by the Experiment Steering strategic pattern, the experiment controller is either local or remote and may feature a graphical user interface (GUI) or some other human-machine interface (HMI).
- The actuator is part of the STEM and moves the scanning electron beam.
- The test is performed in the STEM experiment determines the properties of microscopic structures.
- The sensor is part of the STEM and provides the raw microscope data.

Note that the strategic and architectural patterns present design choices and not specifics about the implementation of the design. The patterns present the overall control, data and work flow, but not the detailed control messages, data transfers and orchestration. For example, the analysis system may be a small, dedicated computer, a larger, shared cluster computer, or an extreme-scale supercomputer. The controller may be a separate dedicated laptop or server, or co-located on the analysis system. Data transfers may be file-based or streaming. Network connections between these components may be dedicated or shared.

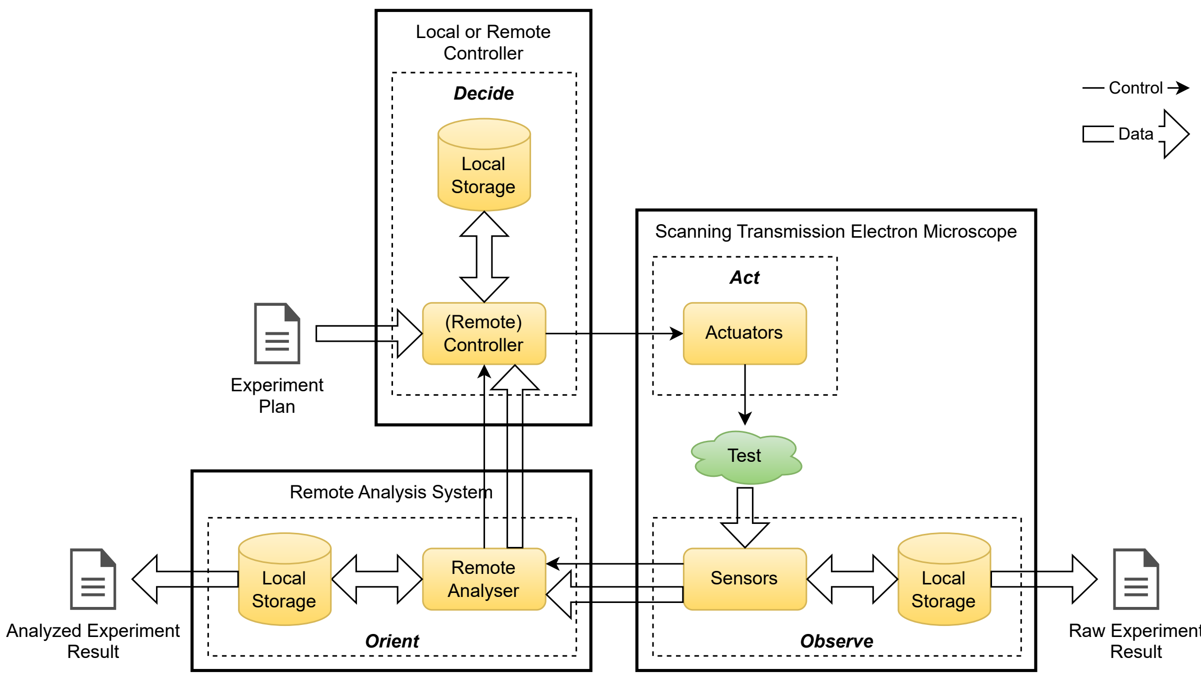


Figure 7-14. Distributed Experiment Steering architectural pattern for the autonomous microscopy science use case

7.5 AUTONOMOUS ROBOTIC CHEMISTRY LABORATORY

Currently, both the rate and output of traditional materials synthesis and discovery are too slow and too small to efficiently provide needed advances. An autonomous robotic chemistry laboratory (ACL) (Figure 7-15) can operate 24/7 with high precision to greatly accelerate materials discovery and innovation. It relies on the design of a laboratory utilizing robotic and autonomous tools for the manipulation of laboratory equipment and characterization tools within the laboratory space. A robotic platform with three major components is used: a mobile base, a robotic arm, and software/characterization tools including integration/feedback with artificial intelligence (AI). To mimic the functionality of a human chemist, each robot action is designed and tested, and the lab equipment modified to be compatible with the robot. Applications focus on the organic synthesis and solid-state synthesis of CO_2 conversion catalysts.

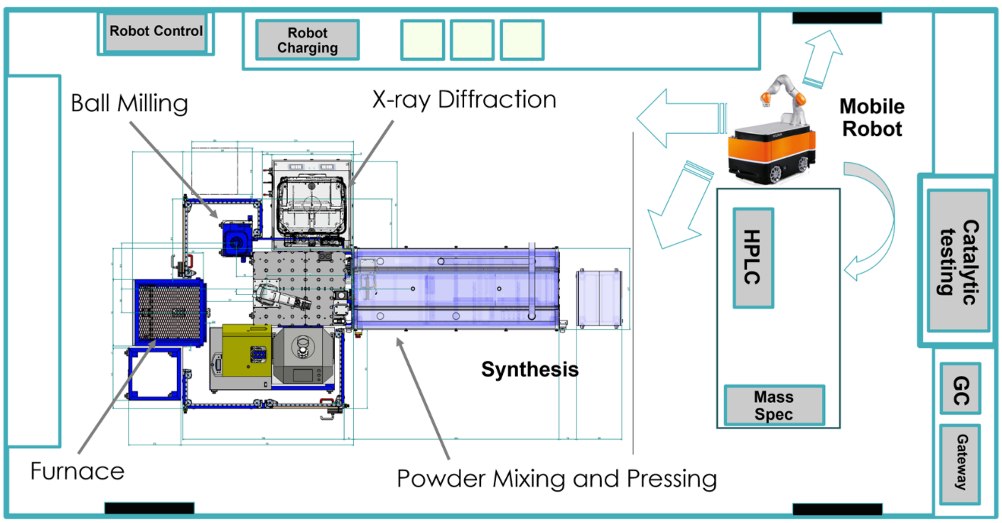


Figure 7-15. The INTERSECT autonomous robotic chemistry laboratory operates 24/7 using analysis of experimental data for the design of experiments.

The ACL science use case implements the Design of Experiments strategic pattern (Figure 7-16), as a robot automates experiment execution and the software/characterization tools in the feedback loop to plan the experiments to be performed. At the strategic pattern level of abstraction, the individual pattern components are as follows:

- The experiment design plan describes the goal, which is the desired chemical compound, and the logic necessary to craft subsequent experiments towards achieving the goal.
- The experiment planner is the SME that may be substituted by a machine learning or deep learning model for autonomous operation to decide on the next experiment plan, given the results from past experiments.
- The experiment plan is the sequence of predetermined steps and associated parameters necessary to run the experiment. The predetermined steps include the parameters for synthesizing the chemical compound, route navigation instructions for the robots to move the sample between the different synthesis and characterization stations, parameters for characterizing the synthesized chemical compound, and safety related feedback instructions.
- The experiment controller is a central workstation that is able to command and control the robots, synthesis equipment, analytical instruments, and any data and computing resources for analyzing the

measurement data.

- The test performed in an experiment characterizes the synthesized chemical compound.
- The experiment result is a combination of the sample characterization results.

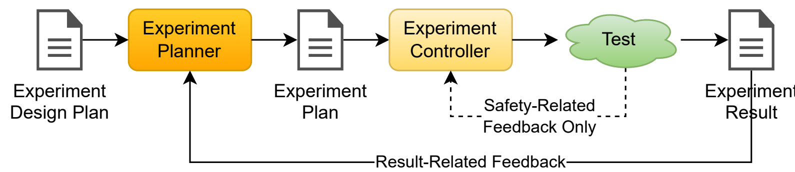


Figure 7-16. Design of Experiments strategic pattern for the ACL science use case

The experiment is a complex sequence of steps involving multiple instruments, actuators, sensors, etc. Thus, the experiment itself could be considered a Multi-Experiment Workflow strategic pattern using a sequence of Experiment Control strategic patterns (Figure 7-17). Examples of steps that constitute the Multi-Experiment Workflow strategic include the synthesis step and each of the individual characterization steps, such as the gas chromatography, high performance liquid chromatography, and X-ray microscopy. Some of these steps could potentially be performed in parallel if the sample were broken down into pieces such that the pieces could be analyzed by the characterization instruments in parallel.

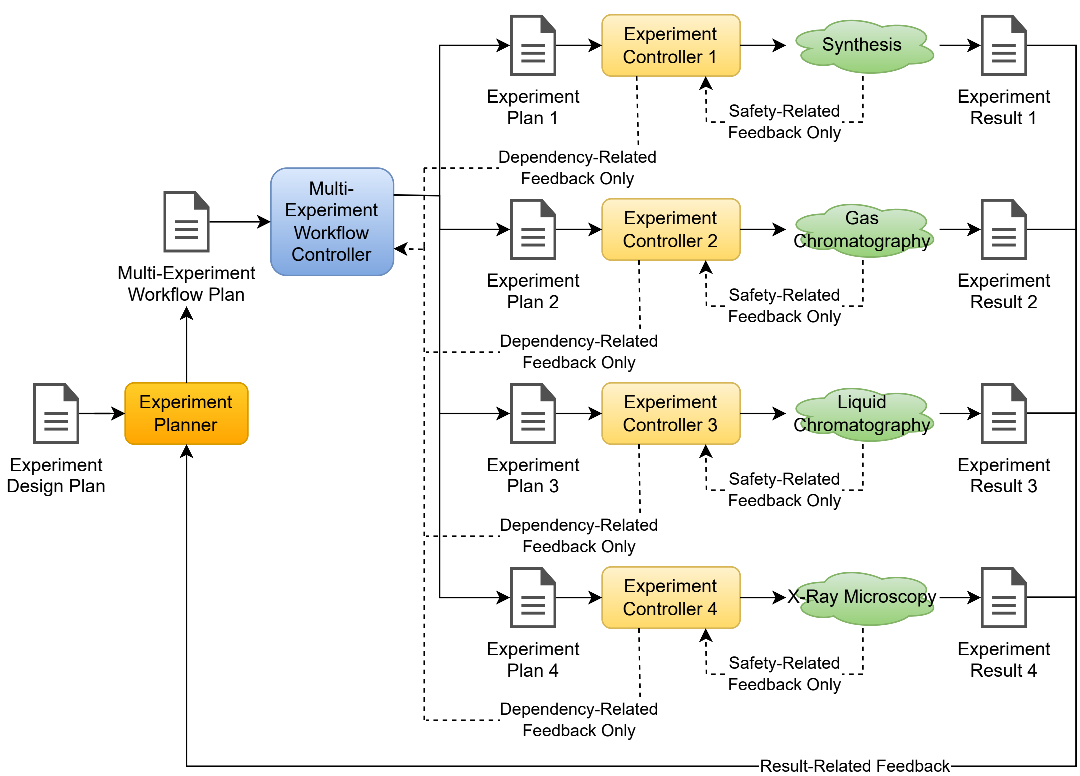


Figure 7-17. Design of Experiments strategic pattern for the ACL science use case, using the Multi-Experiment Workflow strategic pattern.

The ACL science use case implements the Local Design of Experiments architectural pattern (Figure 7-18), as all components (planner, controller(s), robot, synthesis station(s), and characterization station(s)) are local, i.e., in close physical and logical proximity with no significant latency (for communication or sample movement) to remote components. The experiment itself could be considered a Local Multi-Experiment Workflow architectural pattern using a sequence of Local Experiment Control architectural patterns (Figure 7-19). In this case, there is a significant overlap of the different components, as the same shared storage is being used, for example.

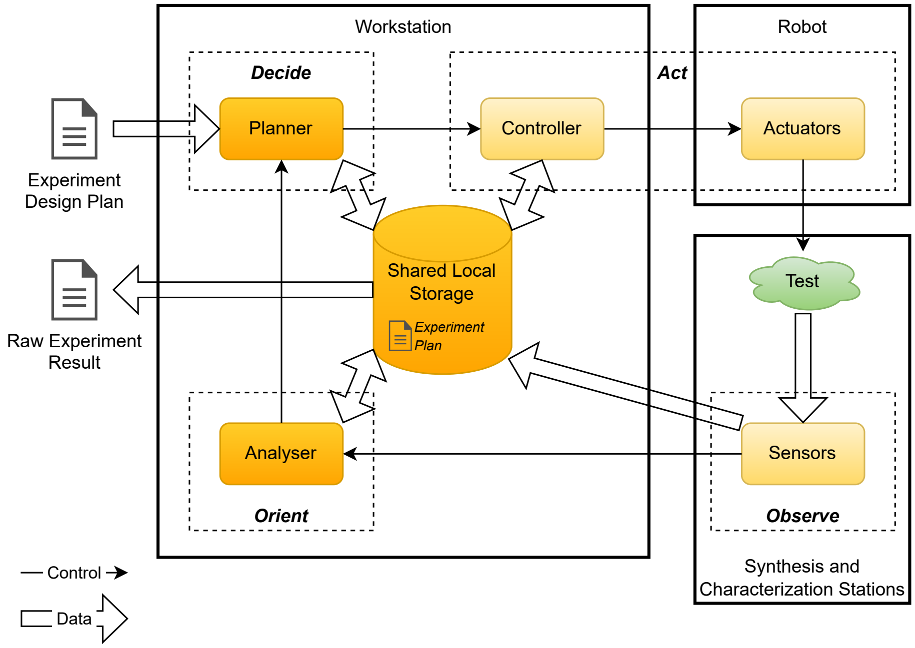


Figure 7-18. Local Design of Experiments architectural pattern for the ACL science use case

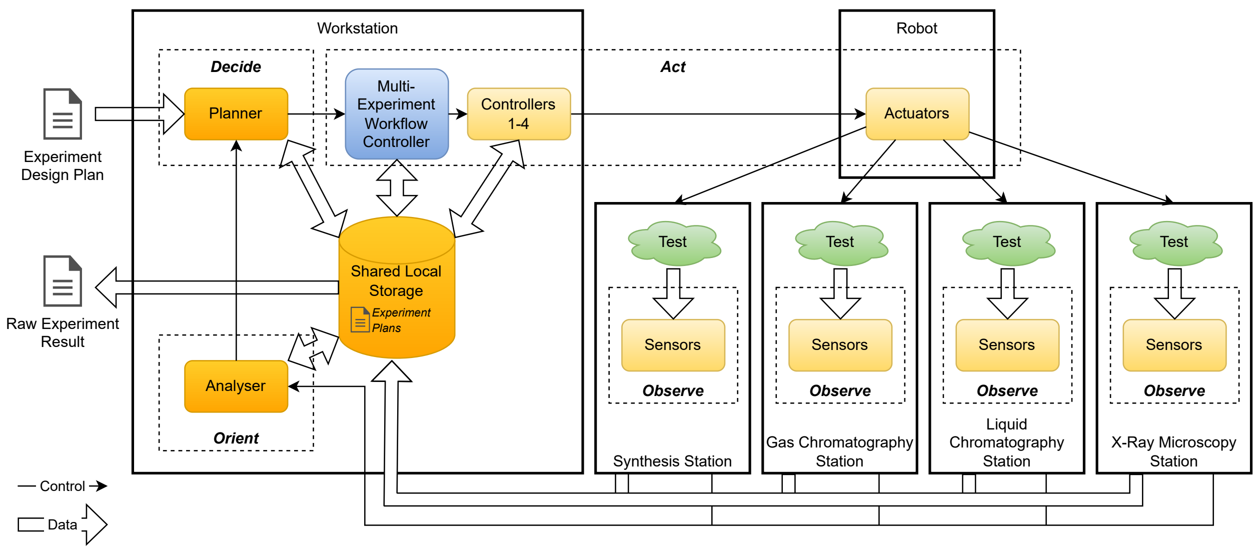


Figure 7-19. Local Design of Experiments architectural pattern for the ACL science use case, using the Local Multi-Experiment Workflow architectural pattern

REFERENCES

[1] DOE national laboratories' computational facilities – Research workshop report. Technical Report ANL/MCS-TM-388, Argonne National Laboratory, Lemont, IL, USA, February 2020. URL https://publications.anl.gov/anlpubs/2020/02/158604.pdf.

[2] Christopher Alexander, Sara Ishikawa, and Murray Silverstein. *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, August 1977. ISBN 978-0195019193.

[3] Jan Borchers. *A Pattern Approach to Interaction Design*. John Wiley & Sons, Inc., New York, NY, USA, 2001. ISBN 978-0471498285.

[4] Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, and Michael Stal. *Pattern-Oriented Software Architecture - Volume 1: A System of Patterns*. Wiley Publishing, August 1996. ISBN 978-0-471-95869-7.

[5] Frank Buschmann, Kevin Henney, and Douglas C. Schmidt. *Pattern-Oriented Software Architecture - Volume 4: A Pattern Language for Distributed Computing*. Wiley Publishing, 2007. ISBN 978-0470059029.

[6] Murray Cole. *Algorithmic Skeletons: Structured Management of Parallel Computation*. MIT Press, Cambridge, MA, USA, 1991. ISBN 978-0273088073.

[7] Murray Cole. Bringing skeletons out of the closet: A pragmatic manifesto for skeletal parallel programming. *Parallel Computing*, 30(3):389–406, March 2004. ISSN 0167-8191. doi: 10.1016/j.parco.2003.12.002. URL https://dx.doi.org/10.1016/j.parco.2003.12.002.

[8] Chad Dougherty, Kirk Sayre, Robert Seacord, David Svoboda, and Kazuya Togashi. Secure design patterns. Technical Report CMU/SEI-2009-TR-010, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 2009. URL https://dx.doi.org/10.1184/R1/6583640.v1.

[9] Douglas K. Van Duyne, James Landay, and Jason I. Hong. *The Design of Sites: Patterns, Principles, and Processes for Crafting a Customer-Centered Web Experience*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002. ISBN 978-0201721492.

[10] Martin Fowler. *Patterns of Enterprise Application Architecture*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002. ISBN 978-0321127426.

[11] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-oriented Software*. Addison-Wesley Professional, 1994. ISBN 978-0201633610.

[12] Daniel Garijo, Pinar Alper, Khalid Belhajjame, Oscar Corcho, Yolanda Gil, and Carole Goble. Common motifs in scientific workflows: An empirical analysis. *Future Generation Computer Systems*, 36:338–351, 2014. ISSN 0167-739X. doi: https://doi.org/10.1016/j.future.2013.09.018. URL https://www.sciencedirect.com/science/article/pii/S0167739X13001970. Special Section: Intelligent Big Data Processing Special Section: Behavior Data Security Issues in Network Information Propagation Special Section: Energy-efficiency in Large Distributed Computing Architectures Special Section: eScience Infrastructure and Applications.

[13] Jeffrey Heer and Maneesh Agrawala. Software design patterns for information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):853–860, September 2006. ISSN 1077-2626. doi: 10.1109/TVCG.2006.178. URL https://dx.doi.org/10.1109/TVCG.2006.178.

[14] Saurabh Hukerikar and Christian Engelmann. Resilience design patterns: A structured approach to resilience at extreme scale (version 1.2). Technical Report ORNL/TM-2017/745, Oak Ridge National Laboratory, Oak Ridge, TN, USA, August 2017. URL https://dx.doi.org/10.2172/1436045.

[15] Saurabh Hukerikar and Christian Engelmann. Resilience design patterns: A structured approach to resilience at extreme scale. *Journal of Supercomputing Frontiers and Innovations (JSFI)*, 4(3):4–42, October 2017. ISSN 2409-6008. doi: 10.14529/jsfi170301. URL https://dx.doi.org/10.14529/jsfi170301.

[16] Sergei V. Kalinin, Maxim Ziatdinov, Jacob Hinkle, Stephen Jesse, Ayana Ghosh, Kyle P. Kelley, Andrew R. Lupini, Bobby G. Sumpter, and Rama K. Vasudevan. Automated and autonomous experiments in electron and scanning probe microscopy. *ACS Nano*, 15(8):12604–12627, 2021. doi: 10.1021/acsnano.1c02104. URL https://doi.org/10.1021/acsnano.1c02104.

[17] Kurt Keutzer and Timothy Mattson. Our pattern language (OPL): A design pattern language for engineering (parallel) software. 2009. URL https://patterns.eecs.berkeley.edu/?page_id=98.

[18] Michael Kircher and Prashant Jain. *Pattern-Oriented Software Architecture, Volume 3: Patterns for Resource Management*. Wiley Publishing, 2004. ISBN 978-0470845257.

[19] Timothy Mattson, Beverly Sanders, and Berna Massingill. *Patterns for Parallel Programming*. Addison-Wesley Professional, first edition, 2004. ISBN 978-0321228116.

[20] Michael McCool, James Reinders, and Arch Robison. *Structured Parallel Programming: Patterns for Efficient Computation*. Elsevier, 2012. ISBN 978-0-12-415993-8.

[21] Michael D. McCool. Structured parallel programming with deterministic patterns. In *Proceedings of the 2nd USENIX Conference on Hot Topics in Parallelism (HotPar) 2010*. USENIX Association, June 2010. URL https://www.usenix.org/legacy/event/hotpar10/tech/full_papers/McCool.pdf.

[22] Douglas C. Schmidt, Michael Stal, Hans Rohnert, and Frank Buschmann. *Pattern-Oriented Software Architecture Volume 2: Patterns for Concurrent and Networked Objects*. Wiley Publishing, 2000. ISBN 978-0471606956.

[23] Rick Stevens, Valerie Taylor, Jeff Nichols, Arthur Barney Maccabe, Katherine Yelick, and David Brown. AI for science report, March 2020. URL https://www.anl.gov/ai-for-science-report.

[24] Jerry Talton, Lingfeng Yang, Ranjitha Kumar, Maxine Lim, Noah Goodman, and Radomír Měch. Learning design patterns with bayesian grammar induction. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST) 2012*, pages 63–74, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1580-7. doi: 10.1145/2380116.2380127. URL https://dx.doi.org/10.1145/2380116.2380127.

[25] Mohammadreza Torkjazi and Ali K. Raz. A taxonomy for system of autonomous systems. In *2022 17th Annual System of Systems Engineering Conference (SOSE)*, pages 198–203, 2022. doi: 10.1109/SOSE55472.2022.9812673.

[26] U.S. Department of Defense. The DoDAF architecture framework version 2.02, August 2010. URL https://dodcio.defense.gov/Library/DoD-Architecture-Framework.

[27] W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros. Workflow patterns.

Distributed and Parallel Databases, 14(1):5–51, 2003. ISSN 1573-7578. doi: 10.1023/A:1022883727209.

[28] Rafael Vescovi, Ryan Chard, Nickolaus D. Saint, Ben Blaiszik, Jim Pruyne, Tekin Bicer, Alex Lavens, Zhengchun Liu, Michael E. Papka, Suresh Narayanan, Nicholas Schwarz, Kyle Chard, and Ian T. Foster. Linking scientific instruments and computation: Patterns, technologies, and experiences. *Patterns*, 3(10):100606, 2022. ISSN 2666-3899. doi: https://doi.org/10.1016/j.patter.2022.100606. URL https://www.sciencedirect.com/science/article/pii/S2666389922002318.

[29] Hanchen Wang, Tianfan Fu, Yuanqi Du, Wenhao Gao, Kexin Huang, Ziming Liu, Payal Chandak, Shengchao Liu, Peter Van Katwyk, Andreea Deac, Anima Anandkumar, Karianne Bergen, Carla P. Gomes, Shirley Ho, Pushmeet Kohli, Joan Lasenby, Jure Leskovec, Tie-Yan Liu, Arjun Manrai, Debora Marks, Bharath Ramsundar, Le Song, Jimeng Sun, Jian Tang, Petar Veličković, Max Welling, Linfeng Zhang, Connor W. Coley, Yoshua Bengio, and Marinka Zitnik. Scientific discovery in the age of artificial intelligence. *Nature*, 620:47–60, August 2023. doi: 10.1038/s41586-023-06221-2. URL https://doi.org/10.1038/s41586-023-06221-2.