

Kamil WOŁOSZYN, Krzysztof PIOTROWSKI

IHP – Leibniz-Institut für innovative Mikroelektronik, Frankfurt (Oder), Germany

AI4U: DATA INTERFACES FOR MODULAR AI APPLICATIONS

The paper presents the AI4U concept, in which data exchange interfaces allow modules in AI applications to be connected in an organized way. Each interface is described by module metadata, such as type, inputs, outputs, constraints, and tags, stored in YAML or JSON format, which makes it possible to select modules and check the correctness of connections before deployment. The paper also describes the planned validation of this approach on the MobiRobAI platform and points to the possibility of transferring the concept to distributed IoT/WSN measurement systems while taking network and energy limitations into account.

AI4U: INTERFEJSY DANYCH W MODUŁOWYCH APLIKACJACH AI

W pracy przedstawiono koncept AI4U, w którym interfejsy wymiany danych umożliwiają uporządkowane łączenie modułów w aplikacjach AI. Interfejs jest opisany metadanymi modułu (typ, wejścia, wyjścia, ograniczenia i tagi) zapisanymi w YAML/JSON, co pozwala dobrać moduły i sprawdzić poprawność połączeń przed wdrożeniem. Opisano planowaną walidację podejścia na platformie MobiRobAI oraz wskazano możliwość przeniesienia koncepcji na rozproszone systemy pomiarowe IoT/WSN z uwzględnieniem ograniczeń sieciowych i energetycznych.

1. INTRODUCTION

In modern solutions that use artificial intelligence, such as robotics, monitoring, and measurement systems, the final solution is rarely just one program. It is usually a set of modules that perform different tasks, such as data collection, processing, running AI models, decision-making, control, communication, and data storage. This division is natural, but it also creates integration problems. Modules are often developed in different environments, based on different assumptions, and they describe information in different ways. As a result, replacing one element, such as a sensor, AI model, or algorithm, may require changes in many parts of the system.

AI4U [1] is a methodological approach to designing modular AI applications. It assumes a process that starts from a functional goal and leads to a complete system made of hardware and software modules. The methodology defines roles, such as the user, integrator, developer, and the AI4U environment, as well as design stages, including requirement description, component selection, application building, and verification before deployment.

This paper focuses on data exchange interfaces between modules in AI4U. An interface is treated as a metadata-based description of a connection, which supports organized module integration, consistency checks, and easier reconfiguration. Section 2 reviews related work, Section 3 presents the main AI4U approach, Section 4 explains metadata and the selection/composition/validation steps, Section 5 describes the planned validation on MobiRobAI, Section 6 discusses the transfer of the proposed approach to distributed measurement systems, and Section 7 concludes.

2. RELATED WORK

The problem of connecting modules using defined inputs and outputs and formal descriptions appears in several areas. In automation, function block networks (e.g., IEC 61499 Standard for distributed Automation) [2] build applications from blocks connected through ports, which supports transfer and reconfiguration because connections are defined at the model level. In robotics, Robot Operating System 2 (ROS 2) [3] uses topics, services, and actions, with interface definitions that support code generation, it mainly covers transport and message formats, while AI4U focuses more on the design stage, especially selecting modules and checking configuration consistency before deployment. In industrial and

measurement systems, Open Platform Communications Unified Architecture (OPC UA) [4] supports interoperability with a communication layer and an information model, and companion specifications help standardize domain data. In Internet of Things (IoT), The Smart Applications REference (SAREF)[5] provides a reference ontology to unify data meaning, it can complement AI4U by adding clearer interpretation on top of interface metadata. In Wireless Sensor Network (WSN), Sens4U [6] connects modules through interfaces and represents the system as a component graph, AI4U follows a similar direction but adds a structured design process and metadata-based selection and validation.

3. AI4U APPROACH: APPLICATION AS A SET OF MODULES

AI4U assumes that an application should be built from modules in a repeatable way, so it can be developed, modified, and moved to other platforms more easily. Modules are treated like building blocks with clearly described inputs, outputs, and limits, and the application is created by connecting them in an organized structure. In AI4U, the application is described on three levels. First, the information flow between modules shows where data comes from and where it is processed. Second, the behavior logic over time, for example a finite state machine (FSM), describes stages of work and reactions to events like missing data or execution errors. Third, a complete configuration stored in YAML (YAML Ain't Markup Language) or JSON (JavaScript Object Notation) lists modules, their parameters, and the links between them, so it can be used for deployment or for comparing variants. This separation helps distinguish what the system should do (logic and structure) from how it is implemented (specific modules and their requirements), therefore, one module can be replaced without rebuilding the whole application if inputs and outputs stay compatible, and basic consistency can be checked early by validating the metadata structure (e.g., with a schema) and matching Outputs to Inputs.

4. METADATA AS A DESCRIPTION OF CONNECTIONS BETWEEN MODULES

In AI4U, modules are not connected by manual code changes. Instead, each module has a short metadata description in YAML/JSON. It includes the module ID and type (e.g., sensor, actuator, ai_model, logic), inputs and outputs (information types), constraints (hardware, software, or performance limits), and optional tags for domain filtering. To keep the meaning clear, the metadata format is planned to use a formal schema (e.g., JSON Schema) that defines required fields and their data types. In practice, modules can be connected when outputs match required inputs and constraints are met, which reduces common integration errors. Fig 1 shows an example YAML module description with these fields.

```

1 id: tomato_detector_v1
2 type: yolo_ai_model
3 inputs: [image]
4 outputs: [bbox, confidence]
5 platform: arm64
6 latency_ms: 100
7 accuracy: 0.91
8 tags: [greenhouse, fruit, object_detection]
```

Fig. 3. Example of a module description in YAML: module identifier and type, declared inputs and outputs, constraints, and tags used for searching and selecting the module.

Rys. 4. Przykład opisu modułu w YAML: identyfikator i typ modułu, deklarowane wejścia i wyjścia, ograniczenia oraz tagi służące do wyszukiwania i doboru modułu

Module selection follows a simple flow from goal to requirements and then to matching components from the library. The user defines the goal, the integrator turns it into technical requirements (sensor type, AI task, expected latency/accuracy, hardware limits), and the AI4U environment searches and filters modules using metadata. Selected modules are connected by matching Outputs to Inputs, and a consistency check verifies that connections are complete and basic constraints are satisfied. Deployment uses generated configuration and supporting files, and later changes (e.g., a new sensor or model) are

handled by updating metadata and running the checks again instead of rebuilding the system from scratch.

5. PLANNED VALIDATION OF THE APPROACH ON THE MOBIROBAI PLATFORM

As a practical example for testing the AI4U approach, the MobiRobAI project has been selected. The platform developed in this project will be used as a test environment for the methods described in the paper under conditions close to a real application. The goal of the project is to build an autonomous robot for harvesting ripe tomatoes in a greenhouse, where the system must work in a changing environment and perform the task in a repeatable way. The planned tasks of the robot include collecting data from a camera, detecting ripe tomatoes with an AI module, planning motion and manipulation, controlling the robotic arm, and repeatedly performing the work cycle while moving along greenhouse rows. This scenario combines perception, reasoning, and execution, which makes it possible to evaluate whether AI4U makes system building and modification easier.

MobiRobAI is a suitable test case because it includes a set of modules with different roles: a data source module (camera), an AI module (detection), a logic module (for example FSM), an execution module (control).

As part of the validation, the following points are planned to be checked:

- whether describing modules with metadata makes it possible to clearly define required inputs, generated outputs, and usage constraints,
- whether this description makes it possible to select matching system elements and detect typical integration problems, such as missing required inputs, incompatible data types, or unmet hardware or time constraints,
- whether changing one element, for example replacing the AI module or the data source, can be done without rebuilding the whole application, but only by updating the description and checking consistency again,
- and whether the design process is repeatable and leads to a clear set of settings needed to run the system.

The MobiRobAI platform will make it possible to compare different configuration variants in practice and to evaluate whether the AI4U approach reduces integration effort and supports easier application development in later iterations.

6. TRANSFER OF THE APPROACH TO IOT/WSN MEASUREMENT SYSTEMS

The mechanism of describing modules by means of metadata and constructing the final solution from selected components is planned to be transferred to distributed measurement systems, as such systems are also built as combinations of interacting elements responsible for data acquisition, communication, processing, analysis, and storage. Due to this structural similarity, they provide an appropriate context for verifying whether the general methodology developed in AI4U is transferable beyond robotic applications. This means that the same metadata-based mechanism can be used to describe modules, select compatible components, compose them into a consistent solution, and verify whether their connections satisfy the required constraints, even though the nature of the modules themselves and the parameters included in their descriptions are different. In distributed measurement systems, these descriptions should cover not only functional properties, but also operational characteristics such as communication conditions, timing requirements, resource limitations, and energy-related aspects.

Thus, the general composition principle known from MobiRobAI, where chains such as camera – AI – logic – control can be constructed, may be transferred to distributed measurement systems in the form of chains such as measurement node – communication – processing – AI – data storage. An additional issue that becomes particularly important in this domain is the semantic consistency of data exchanged

between modules, since in such systems it is necessary not only to define input and output types, but also to ensure consistent interpretation of data meaning across nodes, services, and application layers.

For distributed measurement systems, it may also be important to keep a consistent meaning of exchanged data (for example units, measurement context, and device capabilities). In the basic AI4U workflow this is handled by a controlled type dictionary and metadata fields, so the method remains simple to use. A semantic layer such as SAREF can be added as an optional extension when interoperability across vendors or domains becomes critical, it can be introduced gradually for selected data types.

7. SUMMARY

The paper presents the AI4U approach to data exchange interfaces in modular AI applications, where connections between modules are described by metadata such as ID, Type, Inputs, Outputs, Constraints, and Tags. This description makes it possible to select modules, build the final solution from chosen components, and check the correctness of connections before deployment. The paper also describes the planned validation of the approach on the MobiRobAI platform and points to the transfer of the concept to distributed IoT/WSN measurement systems by extending the metadata with network and operational parameters. The core AI4U validation is based on schema-validated metadata and rule-based connection checks, while semantic annotation (e.g., SAREF) is treated as an optional interoperability extension.

ACKNOWLEDGMENT

This work was supported by the European Regional Development Fund within the INTERREG VI A BB-PL 2021-2027 Programme, project MobiRobAI, grant number BBPL0100046. The funding institutions had no role in the design of the study, the collection, analyses, or interpretation of data, in the writing of the manuscript, or in the decision to publish the results.

REFERENCES

1. Wołoszyn K., Piotrowski K., Turchan K.: *AI4U: Structured Design Process for Modular AI Applications*, in 22. GI/ITG KuVS Fachgespräch Sensornetze: 11. und 12. September 2025, Clausthal-Zellerfeld.
2. IEC 61499, <https://iec61499.com/>, last viewed 23.04.2026.
3. ROS 2, <https://www.ros.org/>, last viewed 23.04.2026.
4. OPC UA, <https://opcfoundation.org/about/opc-technologies/opc-ua/>, last viewed 23.04.2026.
5. SAREF: the Smart Applications REFERENCE ontology, <https://saref.etsi.org/core/v4.1.1/>, last viewed 23.04.2026.
6. Piotrowski K., Peter St.: Sens4U: Wireless Sensor Network Applications for Environment Monitoring Made Easy, in Proc. 4th International Workshop on Software Engineering for Sensor Network Applications (SESENA 2013), 2013.