Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 126 (2024) 384–389

www.elsevier.com/locate/procedia

17th CIRP Conference on Intelligent Computation in Manufacturing Engineering (CIRP ICME ‘23)

Layer-based information model for 5-axis milling processes using a GraphQL schema

Viktor Rudel[a,*], David Wichter[a], Sven Schiller[a], Georg Vinogradov[a], Aleksandra Müller[b], Philipp Ganser[a], Thomas Bergs[a,b]

[a]Fraunhofer Institute for Production Technology (IPT), Steinbachstraße 17, 52074 Aachen, Germany
[b]Laboratory for Machine Tools and Production Engineering (WZL) of RWTH Aachen University, Campus-Boulevard 30, 52074 Aachen, Germany

* Corresponding author. Tel.: +49 241 8904 316; fax: +49 241 8904 316 1. *E-mail address:* viktor.rudel@ipt.fraunhofer.de

Abstract

The view of process-related information in NC-bound manufacturing technologies can range from a perspective of a single cutter location within a considered toolpath to aggregated data which represent the whole operation sequence of a manufacturing process. Data processing pipelines are often individually programmed for the process planning or machining stage addressing an isolated database abstraction layer. This paper proposes a data modeling approach based on a GraphQL schema empowering the user to query planning and actual process data cross-linked with different aggregation layers by a single request.

Keywords: Information model; Data model; Compuer Aided Manufacturing (CAM); GraphQL; Digital twin; Blisk; 5-axis milling; Data processing

1. Introduction

The increasing use of process-related data in CNC machining provides many opportunities to document, analyze and improve cycle times, part quality, production costs and sustainability of products and processes [1]. To describe the digital representation of a 5-axis milling process, data from different sources must be linked, including planning data (target information) generated within the CAD/CAM system, as well as process data (actual information) acquired during the machining process from the numerical control (NC) of the machine tool and external sensors [2]. In the scientific field, this data is often used as input for the development of various process models to describe physical effects in a cause-effect relationship such as toolpath inaccuracies, thermal behavior of the machine tool, material removal and tool engagement, cutting force, process stability, shape deviations of the workpiece and many more [3].

However, a major challenge in data processing is to merge the process-relevant data including all objects and their properties as well as parameters and meta data into a uniform data schema gathered from individual digital systems and process stages. Many scientific publications describe the benefits of a digital shadow based on the chosen case studies, but they do not explain the information model applied. This leads to a lack of context for the origin of the considered data and encouragement for stand-alone solutions for further data processing steps. Against this background, the focus of this research is the development of a layer-based information model for 5-axis milling processes to achieve two main objectives:

(1) Improvement of interoperability between planning and actual process data through formalization and connection of information sources and their interdependencies,
(2) Standardization for data aggregation through a multi-layer approach including the following categories: infinitesimal cutting element, tool slice, feed per tooth tooltip position, toolpath segment, operation, operation sequence.

10.1016/j.procir.2024.08.381

2. State of the Art

The difference between data and information is that data cannot be interpreted without context, whereas information is structured and related to each other [4]. By providing structure, context, and relationships, an information model enables the independent use of data [5]. The essential factor is the mapping of information to a specific data source. In the case of process optimization, data is often processed with calculation algorithms and simulation models to achieve the most detailed process representation possible. Caesar et al [6] subdivide their implemented information model for machining processes into five data categories: (1) workpiece, (2) process, (3) technology, (4) machine tool and (5) tool data. An alternative approach is represented by the STEP-NC data model, which is organized in a hierarchical structure and contains information related to geometry, task, processing and resource description [7].

2.1. Time-independent planning data in machining

Modern product life cycle (PLM) systems already provide the integration of CAD/CAM/CAE and important process meta data in one digital environment [8]. In terms of process verification and simulation the amount of virtual manufacturing solutions [9] extensively expanded in the recent years. Time-independent planning data is coupled with technology models to describe and predict tool engagement conditions, cutting forces, tool wear, surface roughness, part accuracy or surface integrity based on analytical, numerical, empirical or AI-based modeling methods [10]. Additionally, there is a trend to implement the technology models as microservices in a cloud architecture, to scale up the computational performance of the applied algorithms [11]. In a cross-linked cloud environment, models can interact with each other in the same way as the real physical effects in the machining process interact with each other. Furthermore, they are easily accessible for all stakeholders and implemented data processing services.

2.2. Time-dependent process data in machining

High performance cutting processes are characterized by high dynamics. To achieve the required level of detail, the acquisition of time-dependent process data (e.g., axis drive currents, current positions, etc.) at a sufficiently high sampling rate is necessary. In this context, the Input-Process-Output (IPO) cycle of the numerical control (NC) refers to the process by which the controller receives input, processes it, and produces output. In general, NCs have very fast cycle times, typically in the range of milliseconds to microseconds [12]. This is because they need to be able to process large amounts of data quickly and accurately in order to control the movement and operation of the machine tool. However, the exact cycle time will depend on a variety of factors, including the complexity of the control algorithm, the speed of the processing hardware, and the specific application for which the controller is being used [13].

2.2.1. OPC UA

One of the significant advantages of the OPC UA architecture is its ability to model complex data structures with expressive semantics. Communication occurs in a network-like structure between devices, where clients and servers use OPC UA to exchange data [14]. The server provides nodes in an address space, which can be requested by clients. To build the node structure, information models are used, which allow for an interpretable presentation of existing data with metadata and interrelationships between the data. At present, developments take place to standardize the information models of industrial equipment. For instance, VDMA is developing companion specifications for application domains [15] facilitating interoperability between communication participants at the semantic level. The development of integration models for custom information models into existing OPC UA Servers is part of current research activities [16].

2.2.2. Umati

The Universal Machine Tool Interface (umati) is an initiative aimed at creating a standardized communication interface for machine tools in the manufacturing industry. It is a joint project by the German Machine Tool Builders' Association (VDW) and the OPC Foundation [17]. With an interface based on OPC UA, the umati initiative aims to create a common language for machine tools, which would simplify the process of integrating machines into manufacturing processes and enabling more efficient and flexible production processes. Umati has been gaining traction in the industry, with several manufacturers and software providers implementing the standard in their products [18]. Current publications are also based on using umati for acquiring machine tool data [19].

2.2.3. GraphQL

GraphQL is a query language for APIs that was developed by Facebook in 2012 and released publicly in 2015 [20]. It enables developers to retrieve the exact data they need from an API, in a single request. This contrasts with traditional REST APIs, which often require multiple requests to retrieve all the necessary data. GraphQL enables developers to build their own data structures and types on top of OPC UA, which can be easily queried by clients. This allows APIs to evolve over time without breaking client applications. Additionally, GraphQL enables clients to query multiple data sources in a single request, simplifying the process of retrieving data from multiple systems [21].

3. Methodology for layer model development

Scaling problems are present in various domains of practical investigations. The expression "scaling problem" refers to the phenomenon that units being studied exhibit varying characteristics when observed on different scales [22]. As a result, statistical analyses in machining, e.g., building key performance indicators (KPIs), must consider the degree of abstraction.

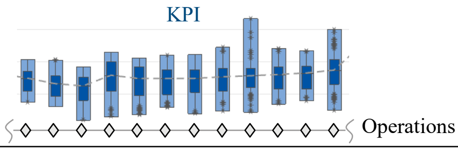
3.1. Formation of a layer structure for machining data

In this context, a layer-based information model for data aggregation (bottom-up direction) of 5-axis milling processes is proposed in Table 1. It contains data analysis approaches, process variables and possible data types for each individual abstraction layer using the example of blisk manufacturing:

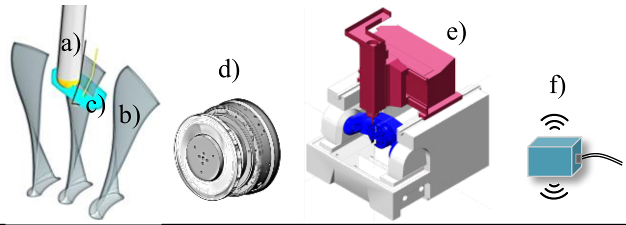
- **Layer 1 - Infinitesimal cutting element (tool slice dz):** Fundamental level of detail to calculate cutter workpiece engagements (CWE) at axial tool location $z_{TCS} = kdz$, slice number, slice height, slice radius, rotational entering and exiting angles ($\varphi_{in}, \varphi_{out}$), axial immersion angle (κ), uncut chip geometry and cutting forces as a function of radial φ and axial κ immersion angles [9]:
  $dF_t = K_{te}dS + K_{tc}h(\varphi_j, \kappa)db$
  $dF_r = K_{re}dS + K_{rc}h(\varphi_j, \kappa)db$
  $dF_a = K_{ae}dS + K_{ac}h(\varphi_j, \kappa)db$
- **Layer 2: Cutter Workpiece Engagement (fz):** Uncut chip geometry and cutting forces integrated along the axial depth of cut and aggregated per move, position interpolation based on feed per tooth fz, immersion angles ($\varphi_{min}, \varphi_{max}, \kappa_{eff}$) per move
- **Layer 3: Tooltip position (TTP)**: Movement type, tooltip position, tool orientation, feedrate, feed vector, workpiece contact point, surface normal vector, e.g.: cutter location source data (CLS) format [23], planned or actual process data mapped to a TTP, etc.
- **Layer 4: Toolpath segment:** Different toolpath segmentation mechanisms are possible:
  a) Toolpath pattern: number of axial/radial cutting feed steps or consecutive TTPs,
  b) Movement Type: Approach, engagement, linking, clearance or retract
  c) Workpiece CAD features, e.g.: suction/ pressure surface, leading/trailing edge, etc.
- **Layer 5: Operation:** Overall information describing a single NC program
  a) Tool system: type, cutting material parameters, diameter, cutting length, helix angle, cantilever length, number of teeth, tool holder properties, etc.
  b) Workpiece system: material parameters, in-process workpiece geometry at start/end, etc.
  c) Planning system: Toolpath, CAM strategy and parameters, target geometry, drive/check curves/surfaces, WCS, etc.
  d) Clamping system: clamping forces, coefficient of friction, support and reference surfaces, etc.
  e) Machine tool system: kinematics, axis limits, precision, zero offset (G54), dynamic settings, NC signals, etc.
  f) Sensor system: type, position, scalar value, timestamp, unit, sensitivity, etc.
- **Layer 6: Operation sequence:** Evaluation of cutting conditions for multiple operations at the same time, e.g.: highest cutting force, tool wear, cumulated machining time, etc.

Table 1: Information model addressing scaling problems in machining

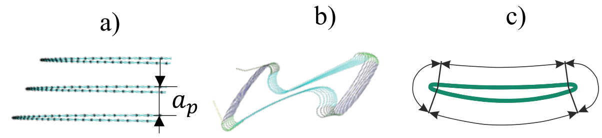
Layer 6: Operation sequence



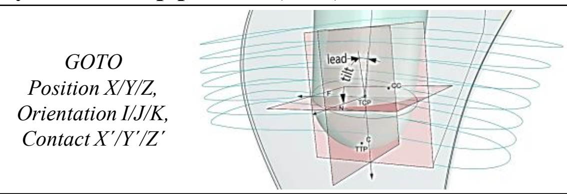
Layer 5: Operation



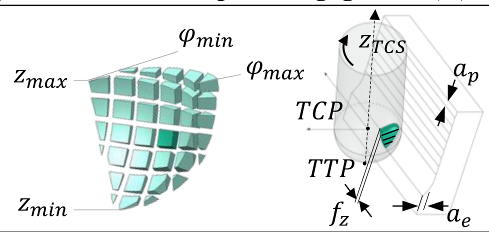
Layer 4: Toolpath segment



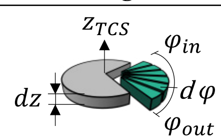
Layer 3: Tooltip position (TTP)



Layer 2: Cutter Workpiece Engagement (fz)



Layer 1: Infinitesimal cutting element (tool slice dz)



3.2. GraphQL Data Schema

As discussed in section 2.2.3, GraphQL can be used to model complex data structures. In the following sections a GraphQL schema capable of incorporating both planning as well as actual process data that is found across all six layers (section 3.1) is proposed. The schema is displayed as a directed graph in Figure 1. In this graph, each type definition of the schema is depicted as a node labeled with the type name. If a type contains a field of another type that is defined in the same schema, an edge is drawn from the former to the latter. Interfaces are shown as nodes with a dashed outline, with implementation relationships between types and interfaces shown as dashed edges. Fields of scalar types are omitted in this depiction. The proposed schema only includes GraphQL queries, as other operations i.e., mutations and subscriptions, are out of the presented schema's scope.

To give the schema the structure of the layer model, each layer is modeled as its own type definition, with the mandatory root type *Query* referencing all six layers. Each layer further contains type definitions, which are either directly included in the type representing the layer or other types belonging to the same layer. Additionally, types can reference other types that are defined on a lower layer. However, a type can never reference another type that is defined on a higher layer than itself. This constraint preserves the hierarchical structure of the proposed layer model and ensures that the level of data aggregation always increases in the direction of lower to higher layers. The schema contains types that relate to planning data, actual data, or both. It is apparent by its name which of these data domains a given type belongs to (and is color-coded Figure 1). The following section describes the types and how they model their respective domain.

3.2.1. Planning and actual data

The planning data contained in the schema is inspired by previous efforts to create a digital twin in blisk manufacturing [2, 5, 11]. The supporting architecture that was built in the wake of these efforts is based on a combination of CAM-based process simulation tasks calculated in a cloud environment. The schema also contains types that model information that is acquired from real, physical processes using technologies such as OPC UA (section 2.2.1). This kind of data is termed actual data.

On the lowest level, the type *InfinitesimalCuttingLayer* models all information as pertaining to the layer of the same name described in 3.1. This consists of information about simulated cutter workpiece engagement on a tool slice level, the latter being modeled by the type *SimulatedCutterSlice*. This type is used for example to calculate differential cutting forces. Information from this layer is integrated one layer above, which is modeled as *CutterWorkpieceEngagementLayer.* Types contained therein, such as *SimulatedEngagement,* hold information about the cutting conditions per cutter move. These engagements are further mapped to a tool position and orientation by being aggregated by the type *PlannedToolMove*, an implementation of the interface *ToolMove*, located on the tooltip position layer. This interface is also implemented by *ActualToolMove*, which represents a monitored executed tool move during the physical process and adds measured data relating to sensors and axes. Many tool moves, either planned or actual, form a tool path segment, as modeled by *ToolPathSegment.* A whole tool path is then given as a collection of many tool path segments and is modeled as such in the interface *Operation*. This interface is the main type on the operation layer and aggregates the relevant machining systems. The two implementations of this interface, *ActualOperation* and *PlannedOperation,* express additional information about an operation (in a way similar to *ToolMove*), with *ActualOperation* adding data about sensor systems and *PlannedOperation* adding planning information such as the CAM strategy. Furthermore, an operation contains various machining systems relevant for its execution, such as the tool, workpiece, planning, fixture, machine tool and sensor system. In order to make these systems additionally available to the client independently from a given operation (for example if the client is interested in the list of all machine tool systems, regardless of operations), these system types are also aggregated by the layer type *OperationLayer* itself. Lastly multiple operations are grouped as sequences in the form of *OperationSequence*, contained in *OperationSequenceLayer*.

4. Benefits and application of a GraphQL server

The currently employed simulation architecture entails that datasets are queried by the user through predefined REST API endpoints [11]. This querying process is limited by the rigid nature of the endpoints: Since the endpoint defines the data set that is to be returned, the client is not able to query the data in a form that is most useful for a given use case. If the client needs to query a different data set, the backend must be adjusted by adding a new endpoint accordingly. Furthermore, in any given data set returned by an endpoint, more data might be contained than is needed by the client, resulting in a wasteful and cluttered exchange between clients and the backend. A GraphQL server alleviates these concerns by offering the advantages of a proper query language. The client can specify exactly which subset of the available data is needed and pass arguments to fields (where supported), further specifying to the server what data is needed. Moreover, a distinguished feature of GraphQL is the possibility to mask multiple data sources behind a unified interface: Even though the client, from its own perspective, only requests the data once from a single source, multiple different data sources might be queried by the GraphQL server itself.

4.1. An exemplary use case: Toolpath comparison of planned and actual process data

To give an example of the benefits of the proposed GraphQL schema a use case taken from a milling process is considered. During the planning stage a target toolpath is generated in a CAM software. However, during the actual milling operation, the tool positions acquired from the NCU deviate from the ideal toolpath, for example due to dynamic limits or thermal expansion of the machine tool. Therefore, the use case of comparing the planned with the actual toolpath presents itself. A client application can query the necessary data for this use case in two steps: First, a list of all operation identifiers is queried from the GraphQL server in order to discover which operations are available for comparison. In this query a field *__typename* can be included to differentiate between instances of *PlanningOperation* and *ActualOperation* and thus identify an instance of each for comparison. In a second step the client can then specifically query more details of these two operations of interest by passing their identifiers as arguments to a field in *OperationLayer*. The result of this second query includes the information of the planned and actual toolpath, which can be further processed by the client, for example by calculating the deviations between the planned and actual tool moves.

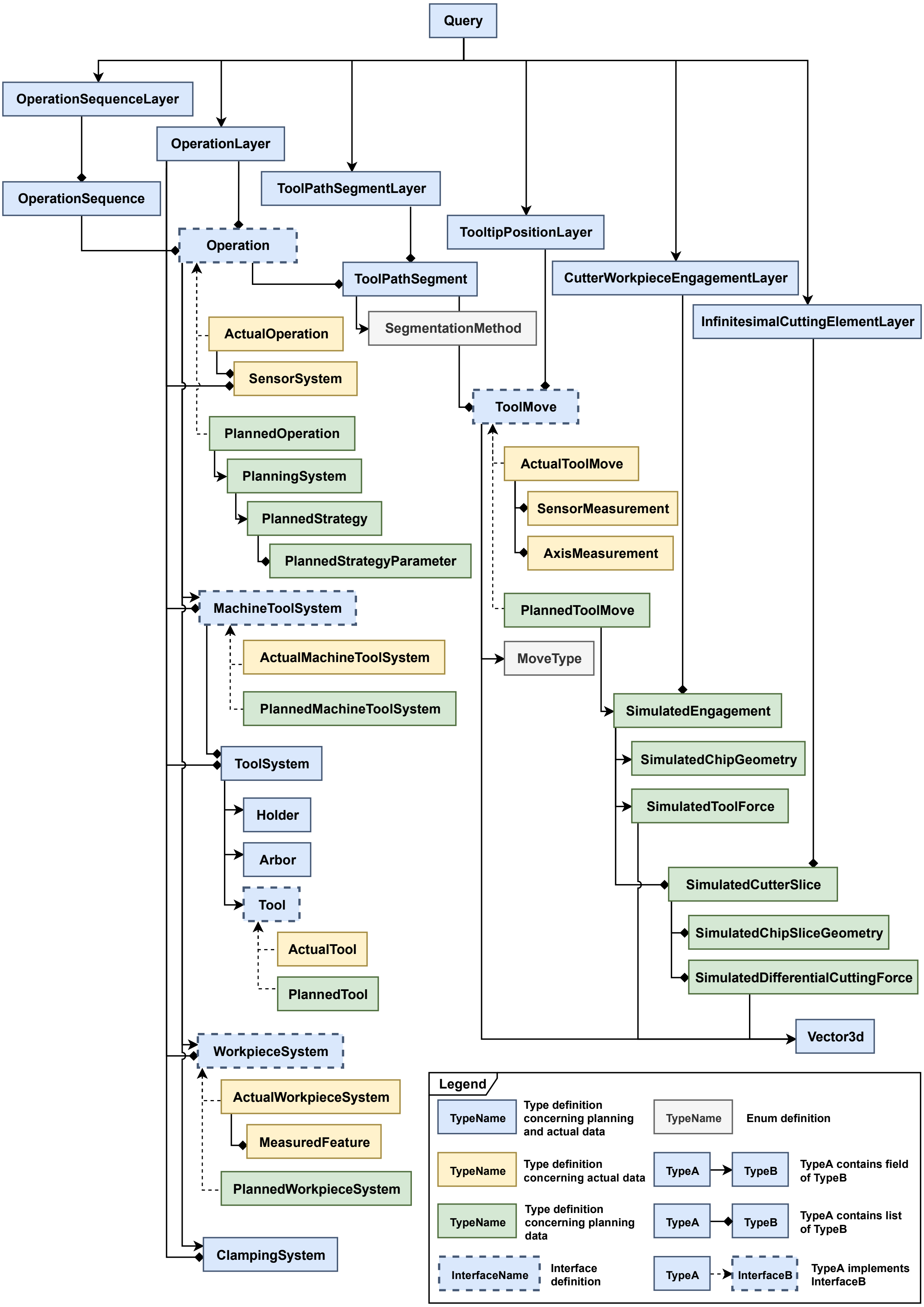


Figure 1: Proposed GraphQL schema for querying planned and actual process data in machining

5. Conclusion

In this paper a model to structure data relating to 5-axis milling processes is proposed. By conceptually differentiating between six layers ordered by their level of aggregation a methodology is shown that enables the unification of planned and actual process data. The model allows for easy extensibility by offering broad categories that are helpful in modeling the heterogenous types of data. As an implementation for the proposed layer model a GraphQL schema is presented that can be used to realize a unified interface for client applications querying machining data. By leveraging the features of GraphQL, problems with previous data model efforts, such as classic REST API's, can be alleviated: through empowering the client to specify exactly what data is needed, hiding the querying of multiple data sources behind a single request, and most of all offering a way to retrieve planning as well as actual data, flexible information processing can be achieved.

Further work in this area is to expand the GraphQL schema to include not only reading access to the data, but also to add support for writing access in the form of mutations. This can for example enable the schema to be usable not only in contexts in which process measurements need to be read but written to a datastore as well. Furthermore, a publication-and-subscription communications model can be realized via GraphQL as well: by adding subscription operations to the schema a client can be kept up to date with new data as it is becoming available to the server. The presented proposal nevertheless shows how value can be added to information processing in manufacturing by unifying data sources through conceptual and implemented information modeling.

Acknowledgements

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Fundation) under Germany's Excellence Strategy – EXC-2023 Internet of Production – 390621612

6. Literatur

[1] Bergs T, Gierlings S, Auerbach T, Klink A, Schraknepper D, Augspurger T. The Concept of Digital Twin and Digital Shadow in Manufacturing. Procedia CIRP. 2021;101:81–4. doi:10.1016/j.procir.2021.02.010.

[2] Ganser P, Venek T, Rudel V, Bergs T. DPART – A DIGITAL TWIN FRAMEWORK FOR THE MACHINING DOMAIN. MM SJ. 2021;2021:5134–41. doi:10.17973/MMSJ.2021_11_2021168.

[3] Hänel A, Seidel A, Frieß U, Teicher U, Wiemer H, Wang D, et al. Digital Twins for High-Tech Machining Applications—A Model-Based Analytics-Ready Approach. JMMP. 2021;5:80. doi:10.3390/jmmp5030080.

[4] VDI. Knowlegde management for engineering: Fundamentals, concepts, approach March 2009;01.140.20, 03.100.99.

[5] Schiller S, Landwehr M, Vinogradov G, Dimitriadis I, Akyürek H, Lipp J, et al. Towards Ontology-based Lifecycle Management in Blisk Manufacturing. Procedia CIRP. 2022;112:280–5. doi:10.1016/j.procir.2022.09.085.

[6] Caesar B, Hanel A, Wenkler E, Corinth C, Ihlenfeldt S, Fay A. Information Model of a Digital Process Twin for Machining Processes. In: 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA); 08.09.2020 - 11.09.2020; Vienna, Austria: IEEE; 2020. p. 1765–1772. doi:10.1109/ETFA46521.2020.9212085.

[7] Zhu W, Hu T, Luo W, Yang Y, Zhang C. A STEP-based machining data model for autonomous process generation of intelligent CNC controller. Int J Adv Manuf Technol. 2018;96:271–85. doi:10.1007/s00170-017-1554-9.

[8] Boschert S, Rosen R. Digital Twin—The Simulation Aspect. In: Hehenberger P, Bradley D, editors. Mechatronic Futures. Cham: Springer International Publishing; 2016. p. 59–74. doi:10.1007/978-3-319-32156-1_5.

[9] Altintas Y, Kersting P, Biermann D, Budak E, Denkena B, Lazoglu I. Virtual process systems for part machining operations. CIRP Annals. 2014;63:585–605. doi:10.1016/j.cirp.2014.05.007.

[10] Arrazola PJ, Özel T, Umbrello D, Davies M, Jawahir IS. Recent advances in modelling of metal machining processes. CIRP Annals. 2013;62:695–718. doi:10.1016/j.cirp.2013.05.006.

[11] Rudel V, Kienast P, Vinogradov G, Ganser P, Bergs T. Cloud-based process design in a digital twin framework with integrated and coupled technology models for blisk milling. Front. Manuf. Technol. 2022. doi:10.3389/fmtec.2022.1021029.

[12] Zhao L, Fang Y, Lou P, Yan J, Xiao A. Cutting Parameter Optimization for Reducing Carbon Emissions Using Digital Twin. Int. J. Precis. Eng. Manuf. 2021;22:933–49. doi:10.1007/s12541-021-00486-1.

[13] Zhang J, Deng C, Zheng P, Xu X, Ma Z. Development of an edge computing-based cyber-physical machine tool. Robotics and Computer-Integrated Manufacturing. 2021;67:102042. doi:10.1016/j.rcim.2020.102042.

[14] Brecher C, Buchsbaum M, Muller A, Schilling K, Obdenbusch M, Staudacher S, Albasatineh MC. Gaining IIoT insights by leveraging ontology-based modelling of raw data and Digital Shadows. In: 2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS); 10.05.2021 - 12.05.2021; Victoria, BC, Canada: IEEE; 10.05.2021 - 12.05.2021. p. 231–236. doi:10.1109/ICPS49255.2021.9468116.

[15] VDMA Mechanical Engineering Industry. Companion specifications 2020. Frankfurt am, Main, Germany.

[16] Muller A, Schnieders T, Storms S, Herfs W. Integration method of custom information models into existing OPC UA Servers. In: 2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA); 06.09.2022 - 09.09.2022; Stuttgart, Germany: IEEE; 2022. p. 1–7. doi:10.1109/ETFA52439.2022.9921670.

[17] Verein Deutscher Werkzeugmaschinenfabriken e.V. umati - universal machine technology interface. 19.04.2023. https://umati.org/industries_machine-tools/. Accessed 19 Apr 2023.

[18] Martins A, Lucas J, Costelha H, Neves C. Developing an OPC UA Server for CNC Machines. Procedia Computer Science. 2021;180:561–70. doi:10.1016/j.procs.2021.01.276.

[19] Kato M, Kizaki T, Uwano T, Iijima K, Kakinuma Y. Development of temperature analysis environment for Cyber-Physical Systems on IoT platform: a study of dynamical properties under temperature change in machine tool spindle unit using carbon fiber reinforced plastics. Procedia CIRP. 2022;107:1485–90. doi:10.1016/j.procir.2022.05.179.

[20] Brito G, Valente MT. REST vs GraphQL: A Controlled Experiment. In: 2020 IEEE International Conference on Software Architecture (ICSA); 16.03.2020 - 20.03.2020; Salvador, Brazil: IEEE; 2020. p. 81–91. doi:10.1109/ICSA47634.2020.00016.

[21] Hietala J, Ala-Laurinaho R, Autiosalo J, Laaki H. GraphQL Interface for OPC UA. In: 2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS); 10.06.2020 - 12.06.2020; Tampere, Finland: IEEE; 2020. p. 149–155. doi:10.1109/ICPS48405.2020.9274754.

[22] Deutsche Forschungsgemeinschaft (DFG). GRK 1644: Scaling Problems in Statistics. 23.04.2023. https://gepris.dfg.de/gepris/projekt/152112243?language=en. Accessed 23 Apr 2023.

[23] Si H, Wang L, Zhang J, Liu Z. A solid-discrete-based method for extracting the cutter-workpiece engagement in five-axis flank milling. Int J Adv Manuf Technol. 2018;94:3641–53. doi:10.1007/s00170-017-1097-0.