2022 IEEE/ACM 26th International Symposium on Distributed Simulation and Real Time Applications (DS-RT) | 978-1-6654-9799-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/DS-RT55542.2022.9932072

# Metadata Specification for DEVS simulation models

Bruno St-Aubin
Department of Systems and Computer Engineering
Carleton University
Ottawa, Canada
bruno.st-aubin@cmail.carleton.ca

Gabriel Wainer
Department of Systems and Computer Engineering
Carleton University
Ottawa, Canada
gwainer@sce.carleton.ca

***Abstract*— Lack of documentation for simulation models impedes their shareability, reusability and composability. There are many organizational challenges that lead to a lack of documentation, and this can have long term consequences: wasted time and effort in the best of cases and irrevocable loss in the worst of cases. A standardized metadata approach is a way to preserve digital resources across time, to support reusability and to mitigate organizational challenges that result in undocumented models. In this paper, we present a metadata specification inspired from the Dublin Core Metadata Initiative (DCMI) with additional elements to capture information specific to Discrete Event System Specification (DEVS) simulation models. This work is meant as an initial step towards the establishment of a metadata standard for the preservation of simulation models.**

***Keywords— DEVS, metadata, discoverability, documentation***

## I. Introduction

In the field of modeling and simulation (M&S), reusability and composability of models are long-standing challenges that have been raised many times by the research community [1]. In practice, models are rarely reused or compatible with a composition approach for many reasons [2]. At the programming level, heterogeneity of programming languages used by simulators and incompatibility between simulation software frameworks prevent reusability. At the design level reusability issues stem from the different paradigms involved in simulation formalisms. At the application level, reusability is more easily achievable since implementation patterns for models can be enforced. But even at the application level, there are issues that can prevent reusability and composition: undocumented or badly documented model components, model components provided as executable binaries, complex source code, etc. [2] Often, modelers will resort to replicating models rather than reusing them outright. To achieve this, modelers must have access to the theory underlying the model.

The discrete event system specification (DEVS) formalism supports reusability at the design level since it provides a formal way of specifying models [3,4]. However, at the application level, it faces the same obstacles as other simulation methodologies. For DEVS, there currently exist no widely accepted documentation pattern for models or other artefacts issued from the simulation process. A thorough and standardized pattern of model documentation can encourage model reuse, composition, and replication by clearly exposing information about a model, its underlying concepts, and important implementation details. Metadata or "data about data" is a common way of documenting digital resources [5]. It organizes a body of knowledge, preserves it for future usage and enables users to explore it efficiently. Yet, in the field of simulation it has been largely overlooked except for some domain-specific applications, such as meteorology [6] or fusion sensor simulation [7].

In this paper we aim to fill that gap by proposing an initial draft of a metadata specification for DEVS simulation models. We first discuss the DEVS specification and existing generic and domain specific metadata initiative. We then describe the metadata specification we propose and discuss concrete ways that it can support the field of modeling and simulation.

## II. Background

### A. Discrete Event System Specification (DEVS)

The Discrete Event System Specifications (DEVS) is a simulation formalism derived from systems theory that was introduced in 1976 by Bernard Zeigler [3]. It has been shown to be a common denominator for other simulation formalisms [4]; it can be used to reimplement models from any other formalism. It is hierarchic and modular. Through composition, any model can be reused as a component to represent larger and more complex systems. DEVS models can be atomic or coupled. Atomic models can be considered as building blocks used to assemble models representing real-world systems. They consist of input and output ports, a persistent state and four behavior functions. Coupled models are structural; they provide a mechanism to assemble atomic models and link them to one another. For a complete description of the formalism, readers should consult [3], the definitive textbook on the DEVS formalism.

To accurately represent the real-world system, domain-specific concepts (physics, biology, sociology, psychology, etc.) must be translated to a DEVS model. This usually requires that subject-matter experts (SME) provide their knowledge about the system to modelers who design and implement models. One way to capture the behavioural and structural elements of a DEVS model is to use a standardized metadata approach. Through metadata, the SME can document the model at a high level before the modeler translates it to a conceptual model and then implements it.

However, there are currently no widely accepted metadata specification for DEVS models.

### B. Metadata and simulation

A standardized metadata approach is a way to preserve digital resources and a tool to mitigate the organizational challenges that lead to dark data. The usefulness of metadata seems obvious but in practice, organizations tend to dismiss or overlook the documentation of resources. Organizations perceive the immediate costs of documenting resources as greater than the future costs of lacking documentation [8].

Metadata promotes findable, accessible, interoperable, and reusable (FAIR) data. They typically document three aspects of a resource: descriptive elements to find and understand resources, structural elements for relationships of the resource with other resources and administrative elements to manage a resource. There are many well-established, generic metadata specifications that extensively document properties and attributes of a resource such as DataCite [9], PREMIS (PREservation Metadata: Implementation Strategies) or the Dublin Core Metadata Initiative (DCMI). There are also well-established metadata standards in specific industry or research fields. In the field of Geographic Information Systems for example, the ISO 19115-1:2014 standard defines a schema for describing geographic data. The Statistical Data and Metadata Exchange (SDMX) standard provides an integrated approach for organizations to manage reporting, exchange and dissemination of statistical data and related metadata. There are numerous such examples in various other fields.

In the field of simulation, metadata standards are uncommon and usually constrained to specific domains of application. For example, the Common Information Model has been proposed to describe climate models [6]. In [7], the authors presents a catalog to document fusion simulation data. It facilitates data location, access, visualization, and analysis. Some research also seeks to leverage existing metadata standards in the context of simulation. The research in [10] introduces a toolkit to extract metadata from simulation output files following the EngMeta format.

## III. A Metadata Specification for DEVS models

The metadata specification we propose is derived from the Dublin Core Metadata Initiative (DCMI) with additional elements to describe DEVS models. Table 1 in appendix describes each element of the specification. For each element, we provide its label, a description and indicate whether they are mandatory (M), optional (O) and repeatable (R). A more detailed version of the specification with examples is available at https://staubibr-stable.github.io/doc-meta. Here, we focus the discussion on the new elements that were introduced to support DEVS and those that expand on the original DCMI elements.

First, we expand on the coverage elements of the standard by separating it into two more precise sub elements: spatial and temporal coverage. Spatial coverage lets users provide either a general named place (e.g. city, province, etc.) or a complete geographic extent ($X_{min}$, $X_{max}$, $Y_{min}$, $Y_{max}$ and a coordinate reference system). The latter is an effort to offer a more machine-readable format for applications that will be discussed in the next section. Similarly, the temporal coverage is more structured; it requires a start date and time as well as a date-time schema (e.g. ISO-8601).

The rest of the specification focuses on DEVS specific elements. They document the structure and behaviour of DEVS atomic or coupled models with the goal of providing a general, high-level understanding of the model. The time metadata element documents the time representation used for the model. In DEVS, it is possible for models to rely on different time representations. For example, one model could consider one unit of time to be a day while another would consider it to be a second. The behavior element provides a high-level description of its internal, external, output and time advance functions. The state element provides information on variables that constitute the state of a model. A state is associated to a message type that it uses when it is output and logged. This allows the automatic reconstruction of output messages when visualizing the trace.

A subcomponent is a model instance that composes a coupled model. Therefore, only coupled models should document this element. Each subcomponent consists of a value that identifies the instance and a model element that identifies the model type associated to the instance. For example, a *processor* coupled model could have 4 instances of a *core* model where each one is identified by a letter. Coupled models should also provide a coupling element that documents the internal and external couplings of the model. Each coupling identifies the origin model (from model), origin port (from port), the destination model (to model) and the destination port (to port) involved. The repeatable port element documents a model's ports through which messages are output. Each port specifies whether it meant for input or output and is associated to the message type used when it outputs and logs a message.

The message elements document each type of messages used by the model, whether to output states or messages through ports. It contains the context required for users to understand the contents of messages. Messages have a unique identifier for unambiguous reference, and they specify field elements that describe their data contents. Each field also has a series of optional qualifiers for the value (name, type, unit of measure). A field element can also contain nested field elements. This is used to represent more message with a more complex data structure.

## IV. Metadata in simulation applications

The model metadata specification is a stable, predictably structured set of information that enables a range of simulation applications. It can serve many purposes: support for the modeling process, the dissemination of simulation results, enhanced model discoverability, preservation of simulation models over time, etc.

### A. Discoverability through libraries of models

Model metadata can be used to build libraries of reusable models that encourage model discoverability, reusability, and their capacity to be shared. Users can find models by querying elements of the metadata specification. For example, the subject element documents a topic for the model

978-1-6654-9799-2/22/$31.00 ©2022 IEEE

through a controlled vocabulary. Users can tag models with relevant keywords: economy, demography, health, energy, disease, network, etc. They can be used to find models applicable to a certain topic. Users can also query spatial coverage to find models that are applicable to a geographic area. By providing the coordinates or the name of a city where their simulation experiment takes place, a spatial intersection can identify all models that are applicable to that geographic area. There are many other, simpler ways of querying a library of models that relies on the specification: models created by a specific author or organization, models created after a specific date, models with keywords in their description, etc. Libraries of models can be exposed through web services and an intuitive web API can be designed to allow easy access to its contents.

### B. *Visual programming and component-based modeling*

Model metadata can enable tools that support users in the modeling process. Visual programming tools like DesignDEVS [11], allow users to assemble models through a graphic user interface (GUI) by dragging and dropping components. Similarly, it can support data-driven component-based modeling methods where simulation models are initialized from data. Both approaches typically require libraries of model to be mapped onto records of a data source representing the real-world system. These approaches have been employed in many application domains: automobile manufacturing, plant engineering, healthcare facilities, planification of constructions operations, and others.

These types of tools can use metadata to provide useful feedback to users. It can help users identify how to map data record attributes onto a model's state. For example, users could create models from rows in a CSV file and map columns to specific state variables. It can also be used to validate that model-components in a model are compatible with one another (by evaluating that time representations for all models match, geographic areas are concurrent, and so on). Messages exchanged through couplings can also be validated according to their contents and units of measure.

### C. *Dissemination of simulation models and results*

In a simulation project, dissemination of results often involves the visualization of traces and analysis of log files. This is generally an ad hoc process; visualizations and analyses are prepared for specific models or experiments. Metadata provides machine-readable information with context that is crucial to the interpretation of the model or results. For example, the description element conveys a high-level understanding of the model to users: equations used by the model, expected logic patterns, interactions with other models, etc. The state element provides insight into the internals of a model; it explains each of the state variables a model uses. Each message used by a model is described by a message element. Field elements provide a field name to associate to message data points. All these information can be shown to users as they are visualizing the simulation trace.

## V. CONCLUSION

Documentation is crucial to the preservation of models and supports the creation of libraries of maodels that can be reused and shared by modelers. A well-defined and machine-readable format provides users with a predictable set of information about a model that can be leveraged in different ways. It enhances model discoverability, supports visual programming or automated component-based modeling approaches, and plays a key role in the dissemination of simulation models and results of simulation experiments. In this paper, we presented a first draft of a metadata specification that we are currently using to design a web-based modeling and simulation environment to support the DEVS simulation lifecycle [12].

We deliberately employed the term specification rather than standard to acknowledge that a full standard for model documentation is a complex undertaking that should be tackled by the broader modeling and simulation community. The Dublin Core metadata standard for example, was proposed in the 1990s and is maintained by the Dublin Core Metadata Initiative. DataCite is the result of a collaboration between researchers of 6 countries since 2009 [9]. Both metadata standards required efforts by a broad community of practitioners collaborating over many years. Only a concerted effort by the simulation research community could lead to a reliable metadata standard for models.

## REFERENCES

[1] S.J.E. Taylor, A. Khan, K.L. Morse, A. Tolk, L. Yilmaz, J. Zander, P.J. Mosterman, Grand challenges for modeling and simulation: simulation everywhere—from cyberinfrastructure to clouds to citizens, Simulation, 91, 2015, pp. 648–665.

[2] O. Balci, J.D. Arthur, W.F. Ormsby, Achieving reusability and composability with a simulation conceptual model, Journal Simulation, 5, 2011, pp. 157–165.

[3] B.P. Zeigler, H. Praehofer, T.G. Kim, Theory of modeling and simulation, 2nd ed. San Diego, CA, USA: Elsevier, 2000.

[4] H.L.M. Vangheluwe, DEVS as a common denominator for multi-formalism hybrid systems modelling, In: CACSD Conference Proceedings, IEEE International Symposium on Computer-Aided Control System Design, 2000, pp. 129–134.

[5] J. Riley, Understanding metadata - what is metadata? Baltimore, MD, 2017, 47 p.

[6] S.A. Callaghan, A. Treshansky, M. Moine, et al., The METAFOR project, In: Proceedings of the 1st International Digital Preservation Interoperability Framework Symposium on - INTL-DPIF '10, New York, New York, USA: ACM Press, 2010, pp. 1–8.

[7] M. Greenwald, T. Fredian, D. Schissel, J. Stillerman, A metadata catalog for organization and systemization of fusion simulation data, Fusion Engineering Design, 87, 2012, pp. 2205–2208.

[8] G. Schymik, K. Corral, D. Schuff, R. St. Louis, The benefits and costs of using metadata to improve enterprise document search, Decision Sciences, 46, 2015, pp. 1049–1075.

[9] J. Neumann, J. Brase, DataCite and DOI names for research data, Journal of Computer-Aided Molecular Design, 28, 2014, pp. 1035–1041.

[10] B. Schembera. Like a rainbow in the dark: metadata annotation for HPC applications in the age of dark data, Journal of Supercomputing, 77, 2021, pp. 8946–8966.

[11] R. Goldstein, S. Breslav, A. Khan, Practical Aspects of the DesignDEVS Simulation Environment, Journal Simulation, 94 (4), 2018, pp. 301–26.

[12] B. St-Aubin, J. Menard, G. Wainer, A web based modeling and simulation environment to support the DEVS simulation lifecycle, In: Proceedings of the 2021 Annual Modeling and Simulation Conference, ANNSIM 2021, pp. 1

978-1-6654-9799-2/22/$31.00 ©2022 IEEE

TABLE I. METADATA SPECIFICATION FOR DEVS ATOMIC AND COUPLED MODELS

| | | |
|---|---|---|
| *Descriptive, rights, technical and structural metadata elements* | | |
| identifier | An unambiguous reference to a resource (e.g. Universally unique identifier) | M |
| title | Name given to the model | M, R |
| alternative | Alternative name to the model title | O, R |
| creator | Person or organization responsible for developing the model. | O, R |
| contributor | Person or organization contributing to the development of the model | O, R |
| type | Identifies whether the DEVS model is “atomic” or “coupled” | M |
| language | Language of the model (i.e. English, French, etc.) | O, R |
| description | A summary description of the model | O, R |
| subject | A topic for the model. | O, R |
| spatial coverage | Spatial coverage for which the model can be used | O, R |
| placename | Name of the location for which the model can be used | O, R |
| extent | Spatial coverage specified by geographic extent | O, R |
| reference | Spatial reference for the spatial coverage (i.e. epsg 4326) | M |
| x min | Minimum x coordinate for the spatial coverage | M |
| x max | Maximum x coordinate for the spatial coverage | M |
| y min | Minimum y coordinate for the spatial coverage | M |
| y max | Maximum y coordinate for the spatial coverage | M |
| temporal coverage | Temporal coverage for which the model can be used | O, R |
| start | Start date or time of the temporal coverage | M |
| end | End date or time of the temporal coverage | M |
| scheme | Date or time schema used to represent the temporal coverage | O |
| license | A link to a legal document giving official permission to do something with the model | O, R |
| created | Date of creation for the model | M |
| modified | Date on which the model was changed | O, R |
| relation | A related resource | O, R |
| *DEVS specific metadata elements* | | |
| time | Time representation used for the model | M |
| behavior | High level description of the model’s behavior | O, R |
| state | State of the model, only applicable for “atomic” models” | O |
| variable | A variable of the model’s state | O, R |
| name | Name of the field, unique within the state object | M |
| description | Description of the field | O, R |
| message | State message type unique identifier | M |
| subcomponent | A subcomponent model instance of the coupled model, only applicable for “coupled” models | O, R |
| identifier | Unique identifier for the subcomponent | M |
| model | Unique identifier reference for the model used for the subcomponent | M |
| coupling | A coupling between an origin and destination model, only applicable for “coupled” models | O, R |
| from model | Origin subcomponent unique identifier for the coupling | M |
| from port | Origin subcomponent port for the coupling | M |
| to model | Destination subcomponent unique identifier for the coupling | M |
| to port | Destination subcomponent port for the coupling | M |
| port | An input or output port of the model | O, R |
| type | Indicates whether the port is an “input” port or an “output” port | M |
| name | Non unique name of the port | M |
| message | Message type unique identifier | M |
| message | Message type used by the model | O, R |
| identifier | Unique identifier for the message type | M |
| field | A field of the message type | M, R |
| name | Name of the field, unique within the message type | M |
| description | Description of the field | O, R |
| type | Indicates whether the field value is “nominal”, “numerical” or “ordinal” | M |
| uom | Unit of measure used for the field, only for “numerical” values | O |
| field | A subfield of the message, defined the same way as the *field* element above | O, R |

978-1-6654-9799-2/22/$31.00 ©2022 IEEE