Supplementary Information

This supplementary information belongs to "Food Safety Knowledge Exchange (FSKX) Format" Software Developer Guide Version 3.2 (https://foodrisklabs.bfr.bund.de/fskx-food-safety-knowledge-exchange-format/).

1. Food Safety Knowledge Simulation Experiment Description Markup Language (FSK-SED-ML)

Food Safety Knowledge Simulation Experiment Description Markup Language (FSK-SED-ML) describes the simulation settings of an FSKX compliant model. FSK-SED-ML extends SED-ML (Level 1 Version 1; for further details see Waltemath, Bergmann, Adams, and Le Novere (2011)), which is based on SBML, by introducing new classes and additional metadata. The main structure of the SED-ML format is maintained (`<listOfModels><listOfSimulations><listOfTasks><listOfDataGenerators><listOfOutputs>`). Note that this document uses the conventions defined in the OMEX specification document (see Bergmann et al. (2014) for details; also see Food Safety Knowledge Exchange (FSKX) Format Software Developer Guide Version 3.2 (https://foodrisklabs.bfr.bund.de/fskx-food-safety-knowledge-exchange-format/) for a summary). In the following, we present the classes used in the FSK-SED-ML. These classes comprise of classes defined by SED-ML (namely Simulation, Model, Task, DataGenerator, Output) and classes defined within FSK-SED-ML (namely SourceScript and FSKSimulationType).

Language References

FSK-SED-ML makes use of an extra file types through the Internet media types, which is previously known as MIME type (Freed & Borenstein, 1996). In Table S1, the URIs for common scripting language is presented.

Table S1 Language references.

Language	MIME type
R	https://iana.org/assignments/mediatypes/text/x-r
Python	https://iana.org/assignments/mediatypes/text/x-python
Matlab	https://iana.org/assignments/mediatypes/text/x-matlab

Class: Model

The **Model** class references the model file that is used for simulations (see Waltemath et al. (2011) for details and the definition). The `language`-attribute is a mandatory attribute. See above sections for definitions and details.

Support for algebraic SBML-based models is also provided. Supported SED-ML classes of SBML are: (1) **Species** for target values, (2) **Parameter** for dependent and constant parameters, and (3) **AssignmentRule** for implementation of equations for target values and dependent parameters. Boundaries for the parameters can be declared by usage of the **Constraint** class. When an already existing SBML-file is used, the SED-ML **Change** class can be used to make modifications. Note that the settings to run a simulation are defined in the class **Task**.

Class: Simulation

The **Simulation** class defines the settings under which models is executed. The **Simulation** class acts as a container for defining the simulation experiments. The **Simulation** class used in FSK-SED-ML is identical to the SED-ML one in Waltemath et al. (2011). In FSK-SED-ML; script-based models use the SED-ML simulation class **SteadyState** whose algorithm has an empty KISAO id (“”). As no algorithm is required within FSK-SED-ML, no KISAO id (algorithm id from the KISAO ontology) is needed.

Simulations are further annotated by **FskSimulationType** and **SourceScript**.

Class: SourceScript

SourceScript is a class that allows to reference external, language-specific scripts (see Figure S1 for the definition). The class contains the following attributes: (1) the path to the scripts (`src` in Figure S1) and (2) the corresponding languages (`language` in Figure S1).

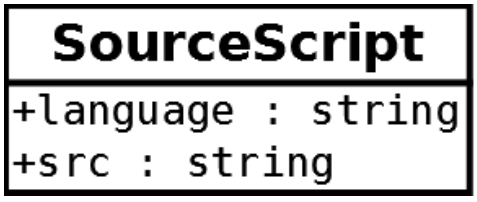

Figure S1 UML class diagram of the SourceScript class.

Attribute: `language`

The required `language`-attribute is of the data type `string`. This attribute is used to specify the scripting language of the code embedded within the **SourceScript** class or referenced through the `src`-attribute (also defined in the **SourceScript** class). Common languages and their MIME types are listed in Table S1.

Attribute: `src`

For the optional source (`src`)-attribute, `anyURI` is used to indicate the path of a local script file. The path is resolved via the local file system or as a relative link. Given this attribute is provided, any free text in **SourceScript** is ignored. When the `src`-attribute is specified in a simulation script, the **Simulation** class should not include script code. If the `src`-attribute is empty, the script needs to be provided within **SourceScript** as free text.

Class: FskSimulationType

FskSimulationType is a class that allows providing metadata about the kind of simulation that will be performed (see Figure S2

Figure S2 UML class diagram of the FskSimulationType class.

for definition). This is an optional class in FSK-SE-ML.

Figure S2 UML class diagram of the FskSimulationType class.

Attribute: `type`

The required `type`-attribute is of the data type `string`. This attribute is used to specify the kind of simulation that will be performed. Allowed values for this attribute are `deterministic`, `statistic`, and `probabilistic` (see Table S2 for definition). At the moment, only `deterministic` type is supported.

Table S2 Definition for the type-attribute values.

Value for the type-attribute	Definition
deterministic	Deterministic simulations operate on predefined values for all model input parameters.
statistic	Statistic simulations create descriptive analysis of observational data. These simulations are meant to describe and analyse data.
probabilistic	Probabilistic simulations use probabilistic methods, like Monte Carlo simulations, to create model input parameters.

Simulations for SBML-based models are defined in this class. Supported SBML-based simulations are: (1) the "Time courses" simulations (only primary and primary-secondary models are supported) and (2) "Parameter scans". Of the predefined SED-ML classes the **SteadyState** is used and expanded by annotations.

Class: Task

The **Task** class defines each simulation scenario through the combination of the classes **Model** with **Simulation** (see Waltemath et al. (2011) for details and the definition). It defines the order in which simulation scenario are executed. If no additional **SourceScript** is defined, the referenced **Simulation**, which uses the default parameter, will be executed.

Each simulation scenario, defined in a **Task,** has to refer to simulation settings defined in a **Simulation**. Nevertheless, it is possible that simulation settings defined as a script can already contain the script command that calls a specific model. In general, a **Task** specifies which simulation setting is combined with which model.

Class: DataGenerator

The **DataGenerator** class defines which values from which simulation scenario is considered for the output (see Waltemath et al. (2011) for details and the definition. A potential reason to process simulation results is to bring values in an appropriate form for later output. It is possible to reference to scripts.

Class: Output

The **Output** class defines how the values specified in the **Task** or **DataGenerator** are plotted (see Waltemath et al. (2011) for details and the definition).

Figure S3 shows an example for a FSK-SED-ML-file. The example is described in Section 7.1 in the main document.

```
<?xml version="1.0" encoding="UTF-8"?>
<sedML  xmlns="http://sed-ml.org/"  xmlns:math="http://www.w3.org/1998/Math/MathML"  level="1"
version="1">
  <!--This file was generated by jlibsedml, version 2.2.3.-->
  <annotation>
    <SelectedSimulation>0</SelectedSimulation>
  </annotation>
  <listOfSimulations>
    <steadyState id="steadyState" name="">
      <annotation>
        <sourceScript              language="https://iana.org/assignments/mediatypes/text/x-r"
src="./param.r" />
      </annotation>
      <algorithm kisaoID=" " />
    </steadyState>
  </listOfSimulations>
  <listOfModels>
    <model                            id="defaultSimulation"                            name=""
language="https://iana.org/assignments/mediatypes/text/x-r" source="./model.r">
      <listOfChanges>
        <changeAttribute newValue="10**rnorm(1000, -1, 1.5)" target="doseValue" />
      </listOfChanges>
    </model>
  </listOfModels>
  <listOfTasks>
    <task           id="task0"           name=""           modelReference="defaultSimulation"
simulationReference="steadyState" />
  </listOfTasks>
  <listOfDataGenerators>
    <dataGenerator id="response" name="">
      <math:math>
        <math:ci>response</math:ci>
      </math:math>
    </dataGenerator>
  </listOfDataGenerators>
  <listOfOutputs>
    <plot2D id="plot1" name="">
      <annotation>
        <sourceScript              language="https://iana.org/assignments/mediatypes/text/x-r"
src="./visualization.r" />
      </annotation>
    </plot2D>
  </listOfOutputs>
</sedML>
```

Figure S3 Example for a full FSK-SED-ML-document.

2. Controlled Vocabularies

Table S3 lists all controlled vocabularies from https://foodrisklabs.bfr.bund.de/rakip-harmonization-resources/ the current version is: https://foodrisklabs.bfr.bund.de/wp-content/uploads/2020/12/Controlled_Vocabularies_Master_Table_V1.04.xlsx (effective 2nd December 2020).

Table S3 Lists all controlled vocabularies from https://foodrisklabs.bfr.bund.de/rakip-harmonization-resources/ (effective 2nd December 2020).

Source	RAKIP project
Rights	Creative Commons
Availability	Knowledge Junction (EFSA) / Zenodo
Format	RAKIP project
Publication Type	RIS format specifications
Publication Status	Bibliographic Ontology Specification
Language	SSD
Software	RAKIP project
Language written in	RAKIP project
Model Class	RAKIP project
Model Sub-Class	RAKIP project + SSD + ICRAd + FDA-iRisk® + PMM-Lab + OpenFSMR (class-specific controlled vocabularies)
Basic process	ICRA + FDA-iRisk® + PMM-Lab (class-specific controlled vocabularies)
Status	RAKIP project
Product-matrix name	SSD
Product-matrix unit	SSD and PMM-Lab
Method of production	SSD
Packaging	SSD
Product treatment	SSD
Country of origin	SSD
Area of origin	SSD
Fisheries area	SSD
Hazard type	SSD
Hazard name	SSD
Hazard unit	SSD and PMM-Lab

Hazard ind-sum	SSD
Population name	FOODON
Region	SSD
Country	SSD
Study Assay Technology Type	SSD
Accreditation procedure Ass.Tec	SSD
Sampling strategy	SSD
Type of sampling program	SSD
Sampling method	SSD
Lot size unit	SSD and PMM-Lab
Sampling point	SSD
Method tool to collect data	RAKIP project
Type of records	RAKIP project
Food descriptors	SSD
Laboratory accreditation	SSD
Laboratory country	SSD
Parameter classification	RAKIP project
Parameter unit	SSD and PMM-Lab
Parameter unit category	PMM-Lab
Parameter data type	RAKIP project
Parameter source	RAKIP project
Parameter subject	RAKIP project
Parameter distribution	probONTO
Model equation class-distr	FDA-iRISK®
Fitting procedure	RAKIP project
Type of exposure	RAKIP project

3. Funding

This work was supported by the EFSA-BfR Framework Partnership Agreement (FPA) GP/EFSA/AMU/2016/01, Specific Agreement Number 3 (SA3) “Establishment of a prototypic QMRA food and feed safety model repository” and the AGINFRA PLUS project funded by the European Commission's Horizon 2020 research and innovation programme under grant agreement No 731001. EMS is funded by the JIP MATRIX within the One Health EJP. One Health EJP has received funding from the European Union Horizon 2020 research and innovation program under grant agreement No 773830.

4. References

Bergmann, F. T., Adams, R., Moodie, S., Cooper, J., Glont, M., Golebiewski, M., . . . Le Novere, N. (2014). COMBINE archive and OMEX format: one file to share all information to reproduce a modeling project. *BMC Bioinformatics, 15*(1), 369. doi:10.1186/s12859-014-0369-z

Freed, N., & Borenstein, N. (1996). Multipurpose internet mail extensions (MIME) part two: Media types. *rfc 2046, November*.

Waltemath, D., Bergmann, F., Adams, R., & Le Novere, N. (2011). Simulation Experiment Description Markup Language (SED-ML) : Level 1 Version 1. *Nature Precedings*. doi:10.1038/npre.2011.5846.1