

SpecIF Introduction

Quick Start Guide

1. SpecIF Goals

SpecIF stands for "Specification Integration Facility". Process and system descriptions from various sources are brought together in an open format and can be published independently of the authoring tools.

The initiators assume that

- there will always be different tools for the disciplines involved;
- It is not sensible to oblige everyone involved to use a certain tool;
- and that there is nevertheless a high level of interest in reading, searching through and checking the work results of all those involved in a common context.

SpecIF is intended to offer a cross-method and cross-manufacturer solution for this purpose. Existing formats and vocabularies are used as far as possible. The motivation of the GfSE working group PLM4MBSE is summarized in a [position paper](#).

Further literature can be found on the [SpecIF homepage](#), including [motivation](#), [examples with online demos](#) and [conference papers](#).

2. Quick Start Guide

SpecIF data can be read with a web browser (preferably Firefox, new Edge or Chrome) using the SpecIF viewer or SpecIF editor. In contrast to a document format, models in SpecIF format can be easily checked because they do not only contain the model elements, but also their semantic relationships can be traced.

Install the SpecIF Viewer/Editor

Install Viewer/Editor on a Web-Server

Download the last released program version "specIF-apps.vX.Y.zip" from <https://github.com/GfSE/SpecIF-Viewer/releases>, to be found in the collapsible "assets" section, and unzip the file in the file system of the web server of your choice. Enter "`http://domain.tld/path/view`" or "`http://domain.tld/path/edit`" in the address line of your web browser to start the respective app. Of course, "`domain.tld/path`" has to be replaced according to your installation.

Use the Installed SpecIF-Viewer/Editor

If the guidelines of your network permit, you can use the viewer/editor, which is installed for demonstration at <https://specif.de/apps/view> or <https://specif.de/apps/edit>. So you always benefit from the latest version.

Prepare the Model-data

Currently the formats SpecIF (JSON), BPMN (BPMN-XML), Archimate (Open-Exchange XML) and MS Excel® (XLSX) are supported.

In case of SpecIF, BPMN and Archimate, no further preparations are required; files of this type can be imported directly.

When importing Excel® files, a folder with the file name as title is created in the hierarchy of the SpecIF data-set. For each worksheet a subfolder is created with it's name as a title. If there is a term in brackets in the name of a worksheet, it is used as the title of the resource class (data type of all resources/objects found on this worksheet). If the worksheet name does not contain a term in brackets, a term in brackets in the file name is used as a substitute. If no type designation is found there either, a generic title is used for the resource class of the worksheet.

Unfortunately, no vocabulary term ready to be interpreted by the system can be used in the filename, as certain characters (e.g. the colon) may not appear in filenames. However, the terms "requirement", "feature", "annotation" and "user-story" are interpreted correctly by the viewer or editor and replaced by a vocabulary term.

SpecIF Introduction

For example, if an Excel file has the file name "Project-name (Requirement) .xlsx" and it contains two worksheets with the names "Functions (User Story)" and "Constraints", a resource-class with the title "User-Story" replaced by "SpecIF:UserStory" and a resource-class with the title "Requirement" replaced by "IREB:Requirement" for the second worksheet are created.

Now for the contents of each worksheet. The property names are expected in the first table line of each worksheet and the resources ("instances") follow from the second line. Defined vocabulary terms are best used as attribute names, e.g. *dcterms:title* and *dcterms:description* from the [Dublin Core Metadata Initiative](#). The use of agreed terms conveys the meaning of the respective attributes so that, just one of the potential benefits, the user interface knows how to display the properties. For example, the title (denoted by *dcterms:title*) comes first in larger letters, underneath the description (denoted by *dcterms:description*) in the main column plus the remaining attributes in a narrower column on the right. However, there are many more benefits in the entire product development.

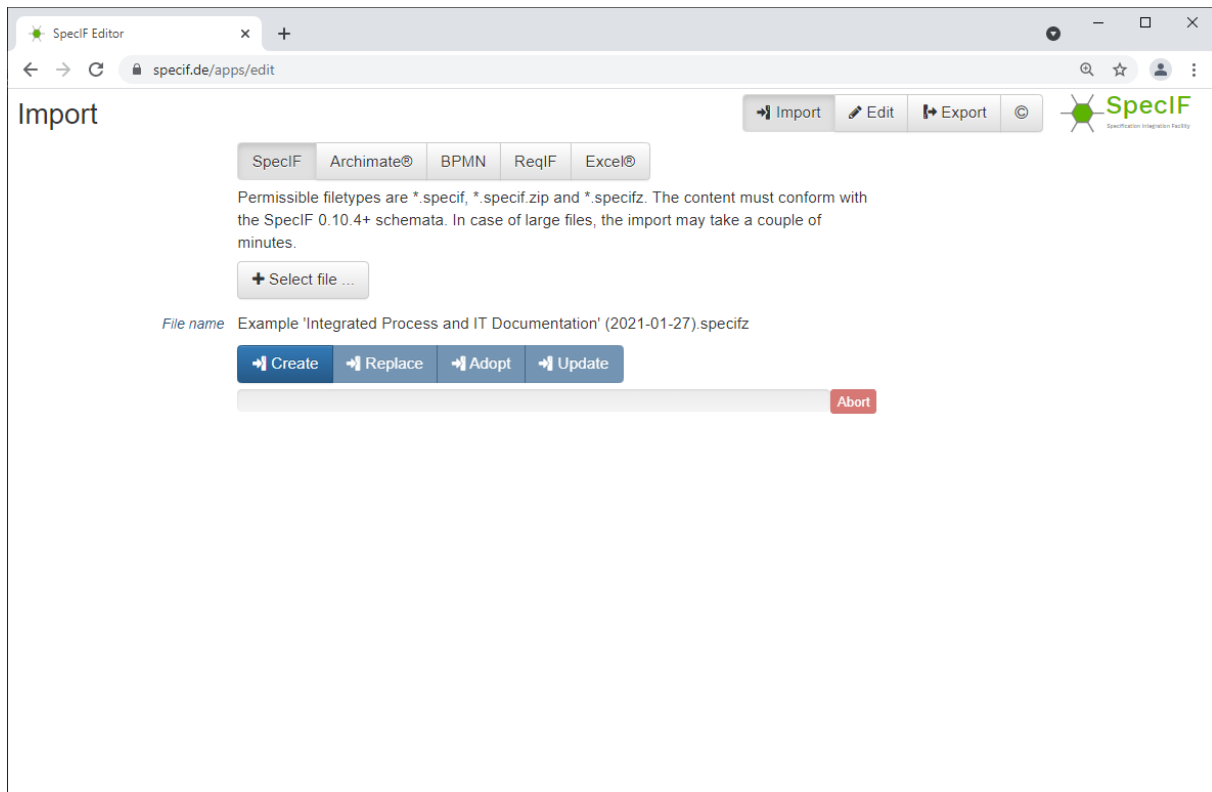
The same goes for the names of relationships; for example *oslc_rm:refinedBy* between two requirements or *oslc_rm:satisfies* between a system component and a requirement. In order to successfully create relationships, the user-defined IDs (*dcterms:identifier*) or titles (*dcterms:title*) of the respective elements must be given between double or single quotation marks. Multiple resources can be specified separated by commas. See columns "IREB:refines" and "oslc_rm:satisfiedBy" in the following figure.

	A	B	C	D	E	F	G	H
1	dcterms:identifier	dcterms:title	dcterms:description	IREB:RequirementType	SpecIF:Priority	IREB:refines	oslc_rm:satisfiedBy	dcterms:modified
2	REQ_1	The system is self-explanatory	80% of the users can use the system right at first contact, thus little training effort is needed.	IREB:QualityRequirement	SpecIF:priorityMedium		"DMS"	2020-03-24
3	REQ_2	Login requires identification	Using the system requires login with identification (authentication). The type of identification is specified in refining requirements.	IREB:FunctionalRequirement	SpecIF:priorityHigh		"FiCo","DMS"	
4	REQ_3	User-identification with user-name and password	The user must correctly enter user-name and password in a login-form.	IREB:FunctionalRequirement	SpecIF:priorityHigh	"REQ_2"		
5	REQ_4	User-identification with finger-print	The user must position a certain finger on a finger-print sensor.	IREB:FunctionalRequirement	SpecIF:priorityMedium	"REQ_2"		

Suitable terms are defined in the [SpecIF Vocabulary](#), namely for [properties](#) and for [statements](#) (relationships).

Import a Model

The start page of the viewer / editor lets you first select the data format and gives you some specific information. Any formats other than SpecIF are converted into the SpecIF format during import.



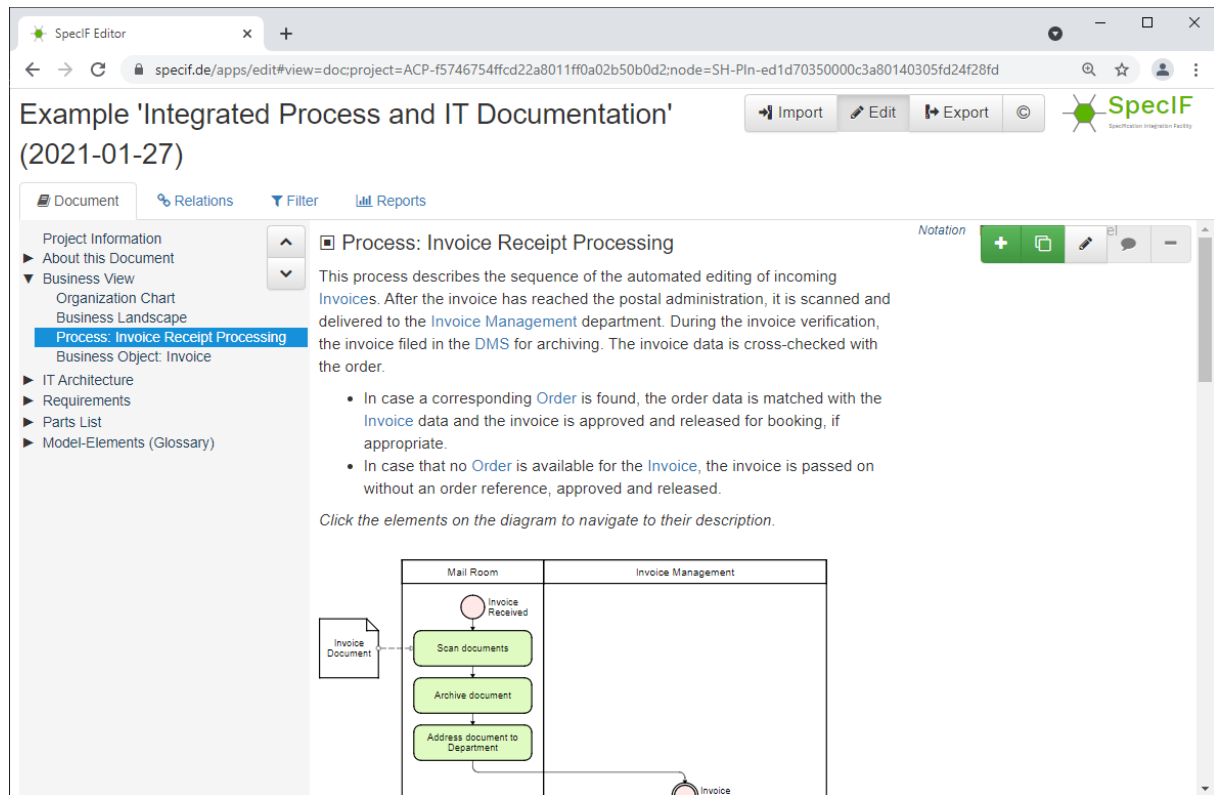
To import a file, after selecting the relevant type, click on "+ Select file" and the familiar file picker of your web browser will open. Select the desired model file and the import will start automatically.

After a successful loading process, the view changes to "Read" resp. „Edit“.

Note: If you load data from your local file system, it will be processed locally by your web browser. The SpecIF viewer or editor does not send any user data over the Internet and does not save any user data on a server. All user data remain local in your web browser.

Read a SpecIF-Model

In the "Document" tab, the model is presented in a structure or outline chosen by the author.



The screenshot shows the SpecIF Editor web application. The browser address bar indicates the URL: `specif.de/apps/edit#view=docproject=ACP-f5746754ffcd22a8011ff0a02b50b0d2;node=SH-Pln-ed1d70350000c3a80140305fd24f28fd`. The page title is "Example 'Integrated Process and IT Documentation' (2021-01-27)". The interface includes a navigation menu on the left with sections like "Project Information", "Business View", "IT Architecture", and "Model-Elements (Glossary)". The "Process: Invoice Receipt Processing" is selected in the outline. The main content area displays the process description: "This process describes the sequence of the automated editing of incoming Invoices. After the invoice has reached the postal administration, it is scanned and delivered to the Invoice Management department. During the invoice verification, the invoice filed in the DMS for archiving. The invoice data is cross-checked with the order." It includes two bullet points:

- In case a corresponding Order is found, the order data is matched with the Invoice data and the invoice is approved and released for booking, if appropriate.
- In case that no Order is available for the Invoice, the invoice is passed on without an order reference, approved and released.

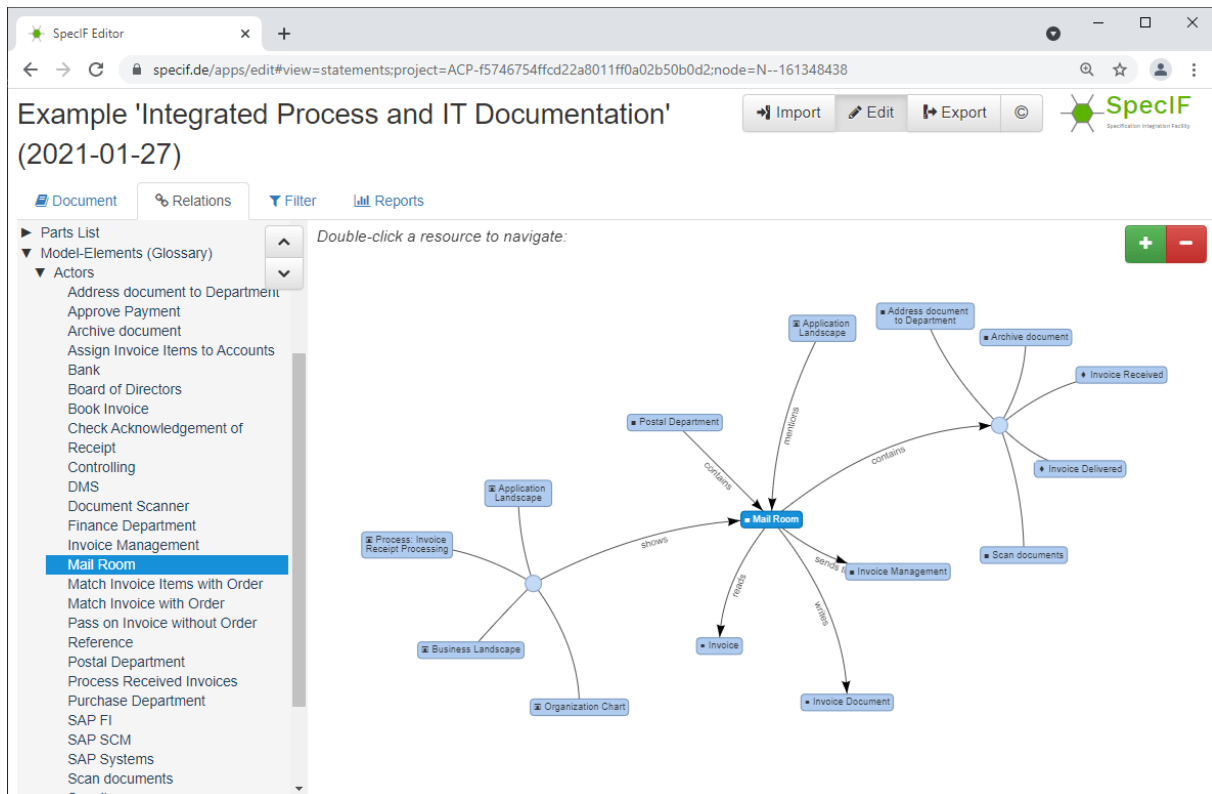
Below the text is a diagram showing a flow from "Mail Room" to "Invoice Management". The "Mail Room" section contains three steps: "Invoice Received", "Scan documents", and "Archive document", followed by "Address document to Department". An "Invoice Document" icon is linked to the "Scan documents" step. The "Invoice Management" section contains an "Invoice" icon. A "Notation" toolbar is visible on the right side of the diagram area.

You can open and close the structural levels on the left; the respective contents are shown on the right. In the main column, the properties designated as title, description and diagram are shown, plus any other properties in a narrow column on the right.

With certain model diagrams, e.g. with FMC- or BPMN-notation, created with [ARCWAY Cockpit](#), the diagram elements are marked such that their title and description are displayed when the mouse is hovered over them. The model element is jumped to with a double click.

Investigate Semantic Relationships

In the "Statements" tab, the semantic relationships of the element selected in the outline to the left are shown. Most relationships are automatically created by graphically analyzing the model diagrams, such as "component XY *contains* function FR" when the block representing function FR is spatially contained in the block representing component XY. Some relationships have to be created manually as part of the analysis and system design, for example "Component XZ *satisfies* requirement 134".



The selected element is a little darker and shown in the center of the spider. This example expresses the following dependencies:

- the "Mail Room" is shown on 4 diagrams,
- it is included in the organization "Postal Department",
- is mentioned by another element „Application Landscape“ in the text,
- itself contains several elements (please double-click the diamond so that the individual elements are shown),
- reads a document „Invoice“,
- writes a document „Invoice Document“ and finally
- sends messages to an organization „Invoice Management“.

The relationships shown are the *entirety* of all statements relating to the selected element in the present model. For example, if one diagram shows that component A is contained in component B and another diagram shows the opposite situation, then it is clearly a contradiction. While it is hardly possible to look through the entire model with all its diagrams and check for consistency, such contradictions are quite easily uncovered when inspecting the statements of individual elements. Current work deals with automatic model testing using hard and empirical rules in order to improve model quality in the long term.

In this view, you can also double-click any element and navigate through the semantic network using the relationships.

License

Both SpecIF as a format with schema and consistency checker as well as the SpecIF viewer/editor are open source and are subject to the [Apache 2.0 license](#), which also allows free use for commercial applications. If used in your own product, the source must be given.

No Guarantee and No Liability

SpecIF and the SpecIF viewer are provided "as-is" and without guaranteed characteristics. Use is always at your own risk. Any liability and guarantee are excluded. The implementations and installations at <http://specif.de> and <https://github.com/GfSE> are reference implementations and currently not designed for high loads and high availability.