

LieberLieber WHITEPAPER

SysMLv2 – Die Zukunft des MBSE

Die finalen OMG-Spezifikationen für SysML v2, KerML und die Systems Modeling API schaffen die Grundlage für die nächste Generation des Model-Based Systems Engineering (MBSE).

Erfahren Sie, welche Neuerungen die Standards mit sich bringen und wie LieberLieber diese Entwicklung bereits mit LemonTree 5.0 unterstützt und Unternehmen bei der Einführung von SysML v2 begleitet.

Die neuen Standards bilden die Grundlage für eine modernisierte, konsistente und interoperable Modellierung – von einem präzisen Fundament (KerML), über die Sprache Systems Modeling Language (SysMLv2) bis hin zur technischen Schnittstelle (Systems Modeling API). Der Schritt war motiviert durch die Einschränkungen der UML, die eine Modellierungssprache für (objektorientierte) Softwaresysteme ist. Außerdem gab es in SysMLv1 Inkonsistenzen in der Semantik und wenig Möglichkeiten für die Weiterverwendung der Modelle bzw. für Automatisierung. Zusätzlich wurde eine API spezifiziert, die es erlaubt, Daten eines SysMLv2 Modells strukturiert abzufragen. Als neues Austauschformat wurde .json festgelegt, was nun endlich zu einer Standardisierung führt und den Austausch zwischen den bisher nicht kompatiblen Systemen in der Modellierung wesentlich erleichtert.

Konzepte der Sprache

SysMLv2 bietet die Möglichkeit folgende Konzepte eines Systems abzubilden:

- Struktur
- Verhalten
- Anforderungen
- Schnittstellen
- Parametrik
- Variabilität

Außerdem beinhaltet die neue Sprache auch die textuelle Notation.

LemonTree – Unterstützung für SysMLv2

LemonTree ist bereits als kommerzielles Diff und Merge Tool global etabliert, allerdings primär für Enterprise Architect UML / SysMLv1 Modelle. Für SysMLv2 unterstützt LemonTree auf Basis des neuen Standards die textuelle Notation (.sysml) bzw. das offizielle Austauschformat .json. Es werden daher alle Modellierungstool unterstützt, die ebenfalls den neuen Standard SysMLv2 umsetzen und daher diese Formate anbieten.

Dazu Dr. Konrad Wieland, Geschäftsführer von LieberLieber: „Wir begrüßen die Einführung von SysMLv2, die uns den Datenaustausch mit vielen weiteren Werkzeugen am Modellierungsmarkt ermöglicht. Allerdings möchten wir betonen, dass SysMLv2 Modelle zwar mit rein textuellen Diff und Merge Tools verarbeitet werden können, aber nur LemonTree sicherstellt, dass das Ergebnis konsistent und valide ist.“

LemonTree liest nämlich den gesamten Modellinhalt ein und interpretiert den vollständigen Abhängigkeitsgraphen. Dadurch können Auswirkungen von Änderungen verständlich angezeigt werden. Das ist speziell notwendig, wenn ein komplexer Sachverhalt in einem SysMLv2 Modell definiert ist und eine Änderung einer Definition (part defs) Auswirkung auf viele davon abgeleitete Verwendungen (parts) hat. Philipp Kalenda, Leiter Consulting bei LieberLieber: „Auch wenn das Modell parallel über eine View und den Texteditor bearbeitet wird, geschieht die Zusammenführung bei LemonTree immer über das zugrundeliegende Modell und nicht über einen textuellen Merge.“

Die erste Preview Version von LemonTree 5.0 wurde Anfang Juni veröffentlicht. Diese Version unterstützt folgende Features:

- Diff und Merge der textuellen Notation (.sysml Files)
- Diff und Merge des SysMLv2 Austausch Formats (.json)
- Migration von .sysml nach .json und umgekehrt

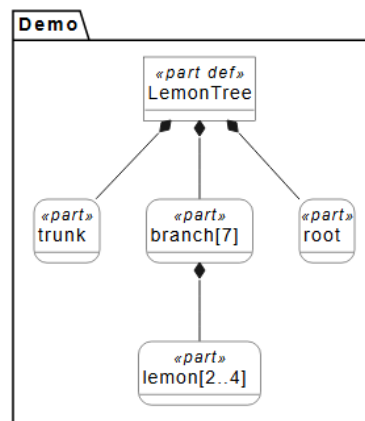
In LemonTree wird es möglich sein Impacted Elements, den Modell Tree sowie die textuelle Notation für die Diff Analyse zu verwenden. Eine Darstellung der grafischen Notation ist bereits in Arbeit.

Textuelle Notation vs. Grafische Notation

SysMLv2 bietet mit der textuellen Notation eine neue Möglichkeit, um Modelle zu erstellen, editieren und visualisieren:

```
part def LemonTree {  
    part trunk;  
    part branch[7] {  
        part lemon[2..4];  
    }  
    part root;  
}
```

Die textuelle Notation ersetzt allerdings nicht die grafische Notation:



Beide Notationen sind gleichwertig und stellen das zugrundeliegende SysMLv2 Modell dar. Toolanbieter verfolgen bei SysMLv2 unterschiedliche Strategien und legen den Fokus auf verschiedene Bearbeitungsmethoden.

Anbieter verfolgen verschiedene Strategien für SysMLv2

Hier eine Auswahl an Anbietern/Open Source Tools, die eine „textual-first Strategy“ verfolgen (Anbieter - Werkzeug):

- Sensmetry – SysIDE
- SysGit – SysGit
- OMG – SysML v2 Pilot Implementation

Im Vergleich dazu eine Auswahl an Anbietern/Open Source Tools die eine „graphical-first Strategy“ verfolgen bzw. bei denen eine (simultane) Bearbeitung beider Notationen für die Zukunft geplant ist:

- Sparx Systems – Enterprise Architect Trechoro
- Dassault Systèmes – CATIA Magic / Cameo SysML v2
- Siemens – Systems Modeler
- IBM – Rhapsody Systems Engineering
- Ansys (Synopsys) – System Architecture Modeler
- Eclipse Foundation – Eclipse SysON

Beide Arten von Werkzeugen bieten darüber hinaus eine Möglichkeit, beide Notationen zu visualisieren, entweder direkt im Tool oder per Export bzw. über die Generierung von Ansichten.

Textuelle und Grafische Notation – Stärken und Zusammenspiel

Die **textuelle Notation** ist ein neuer Zugang zur Modellierung, der es erleichtern soll, Personen aus der klassischen Softwareentwicklung in die Modellierung einzubinden. Außerdem erlaubt die textuelle Bearbeitung folgende Use Cases:

- Generative KI
- Massenbearbeitung (Trace and Replace)
- IntelliSense und Syntaxvalidierung

Die **grafische Notation** knüpft an das ursprüngliche Konzept der „visuellen Modellierung“ an, um komplexe Sachverhalte grafisch darzustellen. Dadurch lassen sich Stakeholder außerhalb der technischen Domäne leichter einbinden. Da die grafische Notation nämlich intuitiver zu lesen ist, wird das Modell leichter als allgemeine Informationsquelle einsetzbar.

SysMLv2 erlaubt eine flexible Arbeitsweise, bei der beide Notationen simultan verwendet und deren jeweilige Stärken miteinander kombiniert werden. Unabhängig von der gewählten Notation werden im Modell-Repository die Elemente, Relationen und Ansichten strukturiert als graphenbasierte Informationen abgelegt. Der Modellspeicher bietet also die Grundlage für die Zusammenarbeit, den Datenaustausch sowie die Evaluierung und Automatisierung des Modells.

Versionskontrolle und SysMLv2

SysMLv2 ermöglicht mit der textuellen Notation und den .sysml Dateien eine Versionskontrolle von SysMLv2 Modellen mit Versionskontrollsystemen wie Git. Hierbei wird jedoch nur eine Ansicht (textuell) des Modells unter Versionskontrolle gestellt und beim Vergleichen, Zusammenführen und der Konflikterkennung berücksichtigt. Die textuelle Notation lässt sich zwar zeilenbasiert interpretieren und vergleichen, dabei werden aber Referenzen, Abhängigkeiten und Querverweise nicht berücksichtigt. Trotz der Ähnlichkeit der textuellen Notation mit herkömmlichem Source Code ist ein SysMLv2 Modell nicht damit gleichzusetzen. Daher wird für die Kollaboration mittels Versionskontrolle ein spezielles Diff- und Merge-Werkzeug (LemonTree) benötigt, das Modelldaten als solche interpretiert und diese in ihrer Graphenstruktur darstellt (siehe Erläuterungen weiter oben im Text).



SYSMLV2: MODELL/TEXT/GRAFISCHE DARSTELLUNG

Aktuell liegt in der Community ein starker Fokus auf der textuellen Notation von SysMLv2-Modellen. Diese stellt jedoch – ebenso wie die grafische Notation bzw. die aus SysMLv1 bekannten Diagramme – lediglich eine bestimmte Darstellung des zugrunde liegenden Modells dar. LemonTree setzt deshalb auf die modellbasierte Ansicht: Die baumartige Darstellung (Graph) wird direkt aus dem zugrunde liegenden SysMLv2-Modell generiert. Auch der Modellvergleich erfolgt auf Basis des eigentlichen Modells und nicht anhand seiner textuellen Repräsentation. Dadurch kann LemonTree Änderungen und Unterschiede auf Modellebene präziser und mit einem höheren Informationsgehalt erfassen und darstellen.

FAQs zu Lemontree SysMLv2



Muss ich sofort auf SysMLv2 umsteigen um LemonTree zu verwenden?



Nein, LemonTree bietet nach wie vor einen Support von UML / SysMLv1 Modellen an.

Kann ich SysMLv2 in Kombination mit Git und LemonTree verwenden?



LemonTree bietet einen 3-Wege Diff und Merge an und kann in gängige Versionskontrollsysteme integriert werden. Bei einem Diff und Merge kann aus dem Git Client LemonTree aufgerufen und SysML(v2) Modell übergeben werden. Beim Merge wird das Ergebnismodell in Git zurückgeschrieben.

Wie können Konflikte aufgelöst werden?



LemonTree führt, wenn möglich einen automatischen Merge von Modellen durch. Bei Überschneidungen von Änderungen (z.B. der Wert des Attributs „Masse“ ändert sich in zwei Branches) zeigt LemonTree diesen Konflikt an. Über den Merge Preview kann der Konflikt bearbeitet und damit eine Entscheidung getroffen werden, welcher Wert in das Merge-Ergebnis übernommen werden soll.

Kann ich herkömmliche Prozesse aus der Software-Entwicklung verwenden?



Da LemonTree rein als Diff und Merge Tool agiert, kann es mit einem beliebigen Git Prozess kombiniert werden. Es kann z.B. mit GitFlow und Pull Request in GitHub gearbeitet werden.

Welche Modellierungstools unterstützt LemonTree?



LemonTree ist bereits als kommerzielles Diff und Merge Tool global etabliert, allerdings primär für Enterprise Architect UML / SysMLv1 Modelle. Für SysMLv2 unterstützt LemonTree auf Basis des neuen Standards die textuelle Notation (.sysml) bzw. das offizielle Austauschformat .json. Es werden daher alle Modellierungstools unterstützt, die ebenfalls den neuen Standard SysMLv2 unterstützen und daher diese Formate anbieten.

Für den Inhalt verantwortlich:



LieberLieber Software GmbH
Gumpendorfer Straße 19
1060 Wien, Österreich
+43 662 90600 2017



Autor
Philipp Kalenda
Leitung Consulting



Textbearbeitung
Rüdiger Maier
Leitung PR