LEHRVERANSTALTUNG – JAHR/SEMESTER

Aufgabe 1: **TITEL DER AUFGABE**

Toolhinweise

Stand: 26.07.2022

Inhaltsverzeichnis

1 Installation und Konfiguration

YAKINDU Statechart Tools (SCT) ist ein Eclipse-Plugin. Da es gelegentlich zu Problemen mit einem nachinstallierten Plugin kommt, empfehlen wir YAKINDU SCT als separates Standalone-Programm zu installieren.

Für die Kombination von YAKINDU SCT mit dem für das gegebenen Validierungsframework wird ein **Java Development Kit (JDK)** mit **mindestens Version 10** benötigt (getestet mit den Versionen 10.0.2, 11.0.2, 12.0.1 und 17.0.2 LTS und 18.0.1.1). In dieser Installationsanleitung wird die neuste Version Java 18.0.1.1 verwendet und YAKINDU SCT 4.0.

1.1 JDK und YAKINDU SCT installieren

1.1.1 Windows 10

1. Java herunterladen (auf der Webseite die .exe-Datei auswählen):
   `https://www.oracle.com/java/technologies/downloads/#jdk18-windows`
2. Java gemäß den Anweisungen installieren.
3. Falls noch nicht automatisch erfolgt, Java zur System-Path-Variable hinzufügen:
   a) In der Suchleiste „Systemumgebungsvariablen bearbeiten“ suchen und öffnen.
   b) Den Button „Umgebungsvariablen“ anklicken.
   c) Die Variable „Path“ im unteren Bereich „Systemvariablen“ auswählen und „Bearbeiten“ klicken.
   d) Falls Java noch nicht eingetragen ist, hier den Pfad der neuen Java-Installation hinzufügen (z.B. `C:\Program Files\Java\jdk-17.0.2\bin`).
4. YAKINDU Statechart Tools Professional für „Windows 64 Bit“ Edition herunterladen:
   `https://www.itemis.com/en/yakindu/state-machine/download-options/`
5. Heruntergeladene .zip-Datei an einem beliebigen Ort entpacken.
6. Nachdem die .zip-Datei entpackt wurde, im entpackten Ordner die Datei `SCT.exe` ausführen.

1.1.2 MacOS X

1. Java herunterladen (auf der Webseite die .dmg-Datei für die entsprechende Architektur auswählen):
   `https://www.oracle.com/java/technologies/downloads/#jdk18-mac`
2. Java gemäß den Anweisungen installieren.
3. YAKINDU Statechart Tools Professional für „macOS X 64 Bit“ Edition herunterladen:
   `https://www.itemis.com/en/yakindu/state-machine/download-options/`

4. Heruntergeladene `YAKINDU SCTPRO.app` in den Ordner Programme verschieben und starten.

1.1.3 Linux

Die Installation auf Linux wurde mit Ubuntu Versionen 18.10, 19.04 und 20.04 getestet. Bei anderen Distributionen ergeben sich ggf. Abweichungen.

Variante 1: JDK über PPA beziehen

1. `sudo add-apt-repository ppa:linuxuprising/java`
2. `sudo apt-get update`
3. `sudo apt-get install oracle-java18-installer`
4. Falls mehrere Java-Versionen installiert sind:
   `sudo apt-get install oracle-java18-set-default`
5. Installation mit `javac -version` prüfen.

Variante 2: JDK direkt herunterladen

1. Java herunterladen:
   `https://www.oracle.com/java/technologies/downloads/#jdk18-linux`
2. Zur manuellen Java-Installation von Java aus einer .tar.gz-Datei diese entpacken und den Anweisungen der beiliegenden `README`-Datei folgen.

Beide Varianten: YAKINDU SCT installieren

1. YAKINDU Statechart Tools Professional für „Linux 64 Bit“ Edition herunterladen:
   `https://www.itemis.com/en/yakindu/state-machine/download-options/`
2. Heruntergeladene .zip-Datei an einem beliebigen Ort entpacken.
3. Nachdem die .zip-Datei entpackt wurde, im entpackten Ordner die Datei „`SCT`“ als ausführbar markieren und starten.

1.2 YAKINDU SCT aktivieren

Für Mitarbeiter und Studierende des Hasso-Plattner-Instituts steht eine akademische Lizenz für die Professional Edition von YAKINDU SCT zur Verfügung. Für die Bearbeitungsdauer vom genügt unter Umständen aber auch die automatisch vorinstallierte 30-Tage Probelizenz.

Akademischen Lizenz beantragten

Auf der folgenden Webseite kann mit Angabe einer Universitäts-Email-Adresse eine kostenlose akademische Lizenz beantragt werden:

```
https://www.itemis.com/en/yakindu/state-machine/licenses#
professional-edition-academic-license
```

In der Regel dauert es 1 Tag bis zur Freigabe. Es kann nach Ablauf der Lizenz mit der selben Email-Adresse eine neue beantragt werden.

Einbinden der akademischen Lizenz

1. YAKINDU SCT zeigt zuerst einen „Welcome Screen“ an. Diesen schließen.
2. Im Menü unter „Window“ das Fenster „Preferences“ öffnen.
3. Dort „YAKINDU Licences“ (vorletzter Punkt) auswählen.
4. Auf „Import License File...“ klicken.
5. Die per E-Mail erhaltene Datei auswählen.
6. Bestätigen und das Einstellungsfenster schließen.

1.3 „PROJEKTNAME“-Projekt einrichten

1.3.1 Projektdaten beziehen

Das „PROJEKTNAME“-Projekt wird über HIER LINK ZUM REPOSITORY bereitgestellt. Es wird empfohlen das Repository zu forken, um stets aktuelle Updates zu erhalten. In dem Repository können zudem Issues gemeldet werden.

1.3.2 „PROJEKTNAME“-Projekt in YAKINDU SCT importieren

Das Repository beinhaltet einen Eclipse-Projektordner, der in YAKINDU SCT importiert werden kann.

1. Den Projektordner (über das Repository oder die .zip-Datei im Moodle) auf der Festplatte speichern.
2. YAKINDU SCT, wie oben beschrieben, starten und den „Welcome Screen“ schließen.
3. Links im „Project Explorer“ einen Rechtsklick machen und „Import“ auswählen.
4. Dort „General“ → „Existing Projects into Workspace“ auswählen.
5. Dann oben rechts auf „Directory“ klicken und Projektordner auswählen.

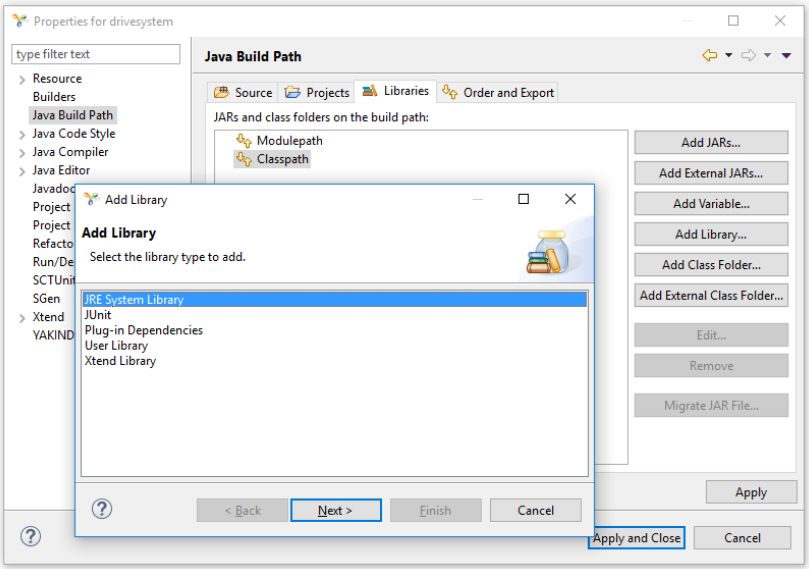


Abbildung 1: In den Projekteinstellungen kann eine bestimmte JRE-Version festgelegt werden

6. In der Projektliste sollte ein Projekt mit dem Namen „mod2-2022-sctsim“ angezeigt werden. Dieses auswählen.
7. Auf „Finish“ klicken, um den Import abzuschließen. Das Projekt ist jetzt links im „Project Explorer“ verfügbar.

1.4 Problembehebung

Das vorgegebene Eclipse-Projekt enthält alle notwendigen Einstellungen und sollte in den meisten Fällen direkt benutzbar sein.

Es kann allerdings passieren, dass lokale Einstellungen (z.B. die installierte Java-Version) Teile der Projektkonfiguration überschreiben. In diesem Fall können die relevanten Einstellungen manuell geändert werden. Dies ist auch notwendig, wenn, anstatt ein Projekt zu importieren (wie im Unterabschnitt 1.3 beschrieben), ein neues Eclipse-Projekt angelegt wird.

Zum Anpassen der Projekteinstellungen kann im „Project Explorer“ das „mod2-2022-sctsim“-Projekt mit einem Rechtsklick angeklickt und „Properties“ ausgewählt werden.

1.4.1 Java-Version manuell festlegen

Ist keine oder eine falsche Version von Java eingebunden, kann dies in den Projekteinstellungen geändert werden.

In den Einstellungen wird dazu „Java Build Path“ ausgewählt und dort der Tab „Libraries“ geöffnet. Mit dem Button „Add Library...“ kann eine „JRE System Library“ hinzugefügt werden (siehe Abbildung 1). Falls in der Liste bereits ein JRE-Eintrag vorkommt, kann dieser per

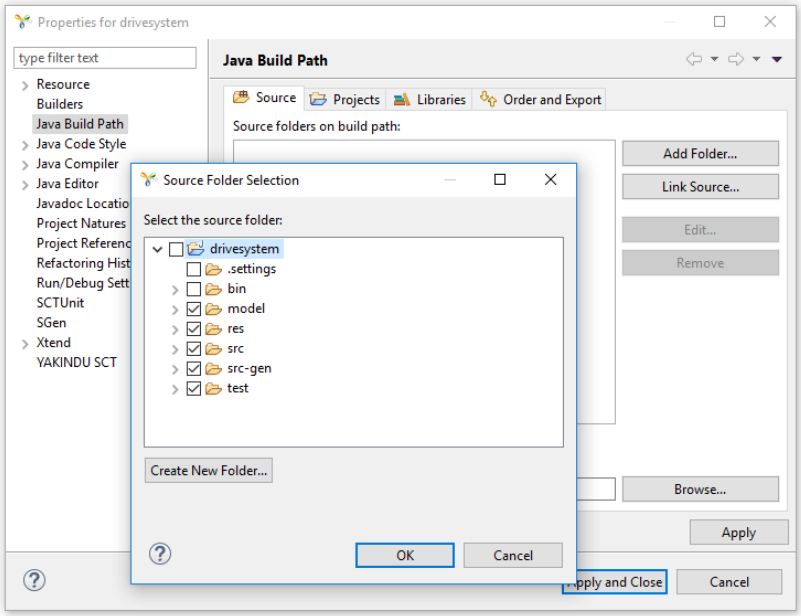


Abbildung 2: In den Projekteinstellungen können die vom Eclipse-Projekt eingebundenen Quelltextordner festgelegt werden

„Edit...“ angepasst werden. Mit einem Klick auf „Installed JREs...“ kann die Liste der Eclipse bekannten Java-Versionen eingesehen und bearbeitet werden.

In manchen Fällen muss neben dem JRE-Eintrag auch noch ein JUnit-Eintrag hinzugefügt werden. Dabei muss JUnit Version 4 ausgewählt werden.

In seltenen Fällen wird nach dem Ändern der Java-Version an einigen Stellen im Code das Keyword `var` weiterhin als Fehler erkannt. Um dies zu vermeiden, bietet Eclipse als Korrekturvorschlag an, die „Project compliance“ auf die neue Java-Version zu setzen. Zum Nutzen eines Korrekturvorschlags kann auf eine der Fehlermeldungen im „Problems“-Fenster oder an der entsprechend markierten Stelle im Code rechts geklickt werden.

1.4.2 Projektordner manuell festlegen

Wenn beim Programmstart Dateien nicht gefunden werden können, könnte es sinnvoll sein, die Liste der eingebundenen Quelltextordner in den Projekteinstellungen zu kontrollieren.

Dazu wird in den Einstellungen der „Java Build Path“ ausgewählt und dort der Tab „Source“ geöffnet. Mit dem Button „Add Folder...“ können nun Ordner hinzugefügt werden (siehe Abbildung 2). Es müssen die Ordner `model`, `res`, `src`, `src-gen` und `test` ausgewählt werden.

1.4.3 Eclipse-Projekt umbenennen

In seltenen Fällen vergibt YAKINDU SCT dem „mod2-2022-sctsim“-Projekt beim Importieren einen anderen Namen (wie z.B. „mod2-2022-sctsim-master“). In diesen Fällen muss das importierte Projekt umbenannt werden, da für die Codegenerierung der Projektname unbedingt „mod2-2022-sctsim“ lauten muss. Zum Umbenennen muss im „Project Explorer“ rechts auf das Projekt geklickt werden und „Refactor“ → „Rename“ ausgewählt werden.

2 YAKINDU Statechart Syntax

Statecharts können die gleichen Konstrukte abbilden wie UML Zustandsdiagramme. YAKINDU SCT zielt auf die Ausführung von Statecharts ab und erzwingt deswegen eine präzisere und striktere Syntax, damit die Ausführung in jedem Fall eindeutig ist.

Hier werden die wichtigsten Besonderheiten vorgestellt. Eine vollständige Spachreferenz befindet sich auf `https://www.itemis.com/en/yakindu/state-machine/documentation/user-guide/sclang_statechart_language_reference`.

2.1 Zustände und Regionen

Statecharts bestehen aus Regionen, die Zustände beinhalten können, die wiederum Regionen beinhalten können, die neue Zustände beinhalten können, und so weiter. Regionen werden dargestellt als hellgraue Rechtecke, Zustände als hellblaue abgerundete Rechtecke. Regionen können wie Zustände Namen haben, die Benennung ist aber optional. Ein Zustand, der Regionen beinhaltet, kann dennoch lokale Reaktionen haben. Wenn innerhalb eines Zustandes mehrere (orthogonale) Regionen sind müssen Sie eine explizite (Ausführungs-)Reihenfolge haben.

Bei UML-Zustandsdiagrammen sind Regionen ebenfalls Teil der Spezifikation und Logik, sind aber nicht Teil der konkreten Syntax und werden daher nicht explizit graphisch mitmodelliert.

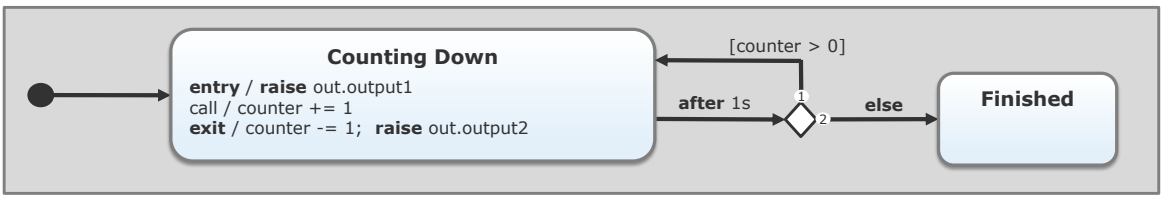
2.2 Reaktionen

Alles Verhalten in einem Statechart wird durch Reaktionen beschreiben, die in Zuständen oder an Transitionen stehen. Eine Reaktion besteht aus einem Trigger (*Eingabe*), eventuell einem Guard (*Bedingung* in eckigen Klammern) und eventuell Effekten (*Ausgaben* und *Seiteneffekte*).

Trigger und Guards funktionieren wie in UML-Zustandsdiagrammen. Zu beachten ist nur, dass „else“ im Statechart ein Trigger statt ein Guard ist und deshalb nicht in eckigen Klammern steht.

Wie bei einem UML-Zustandsdiagramm werden Effekte immer mit einem „/“ vom vorderen Teil der Reaktion getrennt. Mehrere Effekte werden voneinander mit „;“ getrennt. Besonderheit bei den Effekten ist, dass ausgehenden Ereignissen (*Ausgaben*) das Schlüsselwort „raise“ vorangestellt werden muss.

Alle in einem Zustand möglichen Reaktionen haben eine feste Ausführungsreihenfolge. Die internen Reaktionen werden von oben nach unten ausgeführt, die ausgehenden Transitionen aus einem Zustand sind explizit nummeriert, genauso wie die Transitionen die auf eine Verzweigung folgen.



2.3 Definitionsbereich

Ein YAKINDU Statechart hat einen zusätzlichen Definitionsbereich, in dem alle für die Reaktionen benötigten Elemente definiert werden können. Das beinhaltet insbesondere eingehende und ausgehende Ereignisse sowie Operationen, die beispielsweise in Guards aufgerufen werden können. Dies entspricht in etwa dem Ein- und Ausgabealphabet eines erweiterten Automaten bzw. dem Klassendiagramm zu einem UML-Zustandsdiagramm. Für diese Aufgabenstellung ist der entsprechende Teil des Definitionsbereiches fest vorgegeben und **darf nicht verändert werden**.

Außerdem können im Definitionsbereich interne Ereignisse (die per „raise“ aufgerufen und als Trigger empfangen werden können) zur Synchronisation sowie interne Variablen definiert werden. Dieser mit „internal“ markierte Teil des Definitionsbereiches darf für die Lösung von um eigene Ereignisse und Variablen ergänzt werden.

Der Definitionsbereich ermöglicht YAKINDU SCT die strikte Syntaxprüfung bei den Reaktionen.

3 Nutzung von YAKINDU SCT

In diesem Abschnitt werden ergänzend zur Vorstellung der Syntax-Besonderheiten in Abschnitt 2 einige Hinweise zur Nutzung von YAKINDU SCT gegeben.

Eine detaillierte Erläuterung zu YAKINDU SCT findet sich unter `https://www.itemis.com/en/yakindu/state-machine/documentation/user-guide/edit_editing_statecharts`.

3.1 Nutzung von Versionskontrollsystemen

Ein Statechart in YAKINDU SCT ist in einer einzigen .sct-Datei gespeichert. Die .sct-Dateien können *nicht* gemergt werden. Bei der Verwendung eines Versionskontrollsystems (z.B. Git) ist also auf die Vermeidung von Konflikten zu achten.

Wenn parallel auf verschiedenen Geräten an verschiedenen Statechart-Features gearbeitet wird, besteht die Möglichkeit, eine der .sct-Dateien (und die dazugehörige .sgen-Datei) umzubenennen. Die Codegenerierung ist so weiterhin möglich. Zum Zusammenführen der .sct-Dateien ist es in YAKINDU SCT möglich Diagrammelemente zwischen diesen Dateien zu kopieren.

3.2 Modellierung von Statecharts

Das im zu modellierende Statechart befindet sich im Ordner `model`. Per Doppelklick auf die entsprechende Datei `infinitewarehouse_drivesystem.sct` (oder auf eine andere .sct-Datei) öffnet sich die Modellierungsansicht.

Sollte sich die Modellierungansicht nicht wie erwartet öffnen, muss die aktuell gewählte Perspektive überprüft werden (kleine Icons oben rechts). Hier ist „SC Modeling“ auszuwählen.

3.2.1 Überblick

Die Modellierungsansicht (siehe Abbildung 3) ist wie folgt strukturiert:

- In der Bildschirmmitte befindet sich das eigentliche Statechart-Diagramm.
- Auf der linken Seite befindet sich der Definitionsbereich des Statecharts, der dem Klassendiagramm aus dem Entwurfsdokument (siehe Abbildung 5 dort) entspricht. Dieser Definitionsbereich darf nicht verändert werden und dient lediglich als Referenz. Interne Ereignisse und Variablen (gekennzeichnet durch das Schlüsselwort „internal“) dürfen allerdings ergänzt werden.
- Auf der rechten Seite befindet sich die Palette mit den Diagrammelementen, die zur Modellierung genutzt werden können. Besonders wichtig sind hierbei die Elemente „State“ (um einen neuen Zustand anzulegen), „Transition“ (um eine Transition zwischen zwei Elementen einzufügen) und „Choice“ (um eine Entscheidung zu definieren).
- Im unteren Bereich befindet sich das „Properties“-Fenster, in dem Eigenschaften des aktuell ausgewählten Elements zu finden sind.

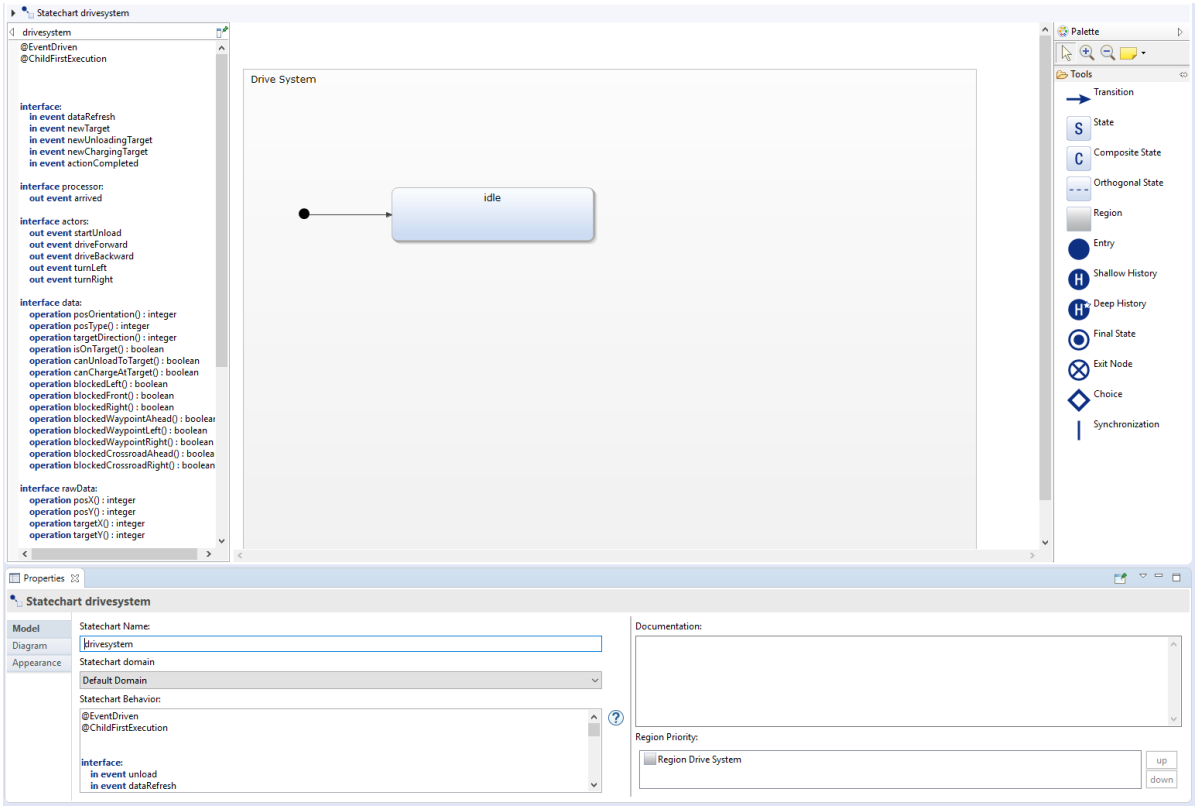


Abbildung 3: Die Modellierungsansicht in YAKINDU SCT

3.2.2 Syntax und Konventionen

Für das Statechart ist die aus der Vorlesung bekannte Syntax zu verwenden.

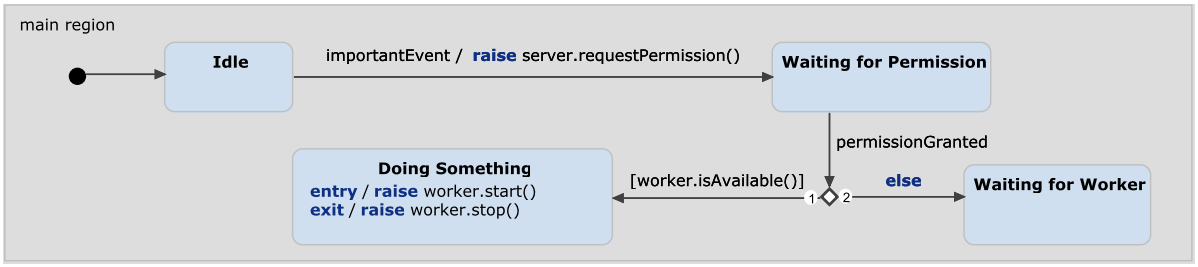
Zusätzlich gelten einige Konventionen bei der Benennung der Zustände und Regionen. Das sollte dem Framework ermöglichen den aktuellen Zustand des Statecharts anzuzeigen:

- Die äußerste Region muss „`Drive System`“ heißen.
- Wenn bei der Modellierung (abgesehen von der äußersten Region) weitere Regionen verwendet werden, müssen diese entweder unbenannt sein oder deren Namen müssen mit einem Unterstrich („_“) beginnen. Die Regionnamen dürfen keine Leerzeichen und keine weiteren Unterstriche enthalten.
- In Zustandsnamen dürfen keine Unterstriche benutzt werden.

3.2.3 Modellierung von Reaktionen

Mittels Doppelklick auf den unteren Bereich eines Zustandes öffnet sich ein Textfeld, in dem lokale Reaktionen (z.B. mit „entry“ oder „exit“ als Trigger) spezifiziert werden können. Mittels Doppelklick auf eine Transition öffnet sich ein Textfeld, in dem auch eine Reaktion spezifiziert werden kann.

Die Trigger einer Reaktion können jeweils direkt angegeben werden. Nach den Triggern können Guards in eckigen Klammern definiert werden. Innerhalb eines Guards können Operationen und Variablen aus dem Definitionsbereich verwendet werden.



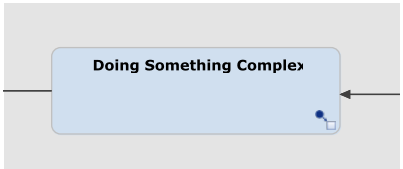
Die Effekte sind durch ein „/“ von Triggern und Guards zu trennen. Hier können unter anderem ausgehende oder interne Ereignisse genutzt werden, die mit dem Schlüsselwort „raise“ eingeleitet werden.

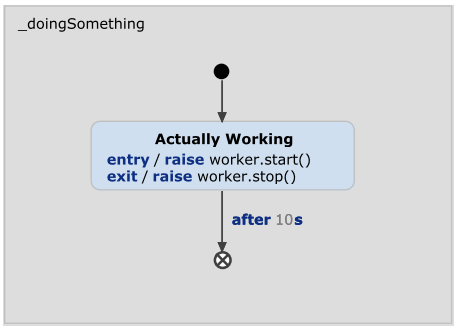
Aus einer Entscheidung oder einem Zustand ausgehende Transitionen sind jeweils nummeriert. Die Zahlen geben dabei die Reihenfolge an, in der die Transitionen ausgewertet werden. Es wird also zuerst die Transition mit der Priorität 1 ausgewertet. Wenn die Transition eine Bedingung hat und diese Bedingung wahr ist, werden die Effekte dieser Transition ausgeführt, ansonsten wird die Transition mit der nächsthöchsten Priorität ausgewertet. Die Prioritäten der ausgehenden Transitionen können im „Properties“-Fenster einer Entscheidung oder eines Zustandes angepasst werden.

3.2.4 Subdiagramme

Um das Statechart übersichtlich zu halten, ist es möglich Subdiagramme anzulegen. Dazu kann ein beliebiger Zustand rechts angeklickt und dann „Create Subdiagramm“ ausgewählt werden, um den Zustand mit einem Subdiagramm zu versehen. Solcher Zustand ist dann mit einem kleinen Icon unten rechts markiert. Statt eines Zustandes mit Subdiagramm kann auch ein hierarchischer Zustand genutzt werden.

Ein Zustand mit Subdiagramm bzw. ein hierarchischer Zustand muss genau eine ausgehende Transition ohne Trigger haben.





Durch einen Klick auf das Icon in der rechten unteren Ecke eines solchen Zustandes wird das Subdiagramm in einem neuen Tab geöffnet. Hier muss zuerst eine neue Region angelegt werden. Innerhalb dieser Region kann nun frei modelliert werden (insbesondere können mehrere Subdiagramme ineinander geschachtelt werden). Dabei muss beachtet werden, dass es innerhalb der angelegten Region ein „Entry“ sowie einen „Exit Node“ geben muss (nicht zu verwechseln mit dem „Final State“). Nachdem das Subdiagramm diesen „Exit Node“ erreicht hat, wird die Behandlung des äußeren Diagramms entlang der triggerlosen Kante fortgesetzt.

3.3 Simulation

Das Simulationsfeature von YAKINDU SCT ermöglicht ein manuelles Simulieren und Testen des Statecharts. Man startet die Simulation für die .sct-Datei durch einen Klick auf den „Run“-Button.

Der Modellierungsbereich wird während einer Simulation durch eine Ansicht ersetzt, in der das Statechart nicht mehr bearbeitet werden kann, aber der aktuelle Zustand bzw. die aktuellen Zustände des Statecharts gelb markiert sind. Im Simulationsmenü (Tab „Simulation“ rechts) können Trigger ausgelöst sowie die Rückgabewerte von Operationen gesetzt werden.

Eine laufende Simulation kann durch einen Klick auf das rote Rechteck („Stop“-Symbol) oben im Simulationsmenü beendet werden.

3.4 Codegenerierung

Neben der internen Simulation bietet YAKINDU SCT die Möglichkeit, Code aus einem Statechart zu generieren. Dies ermöglicht dann das Einbinden dieses Codes in eine eigene Simulationsumgebung (z.B. in den im Abschnitt 4 beschriebenen Validierungsframework).

Zur Codegenerierung muss auf die mitgelieferte Konfigurationsdatei `infinitewarehouse_drivesystem.sgen` rechts geklickt und „Generate Code Artifacts“ ausgewählt werden. Nach der Erfolgsmeldung im Konsolenfenster befindet sich der generierte Quelltext im Ordner `src-gen`.

3.5 Bekannte Probleme

In seltenen Fällen reagiert die Modellierungsansicht von YAKINDU SCT nicht mehr auf alle Mausklicks und Tastendrücke (z.B. auf die „Entf“-Taste). Um dies zu beheben, genügt es die .sct-Datei zu schließen und erneut zu öffnen.

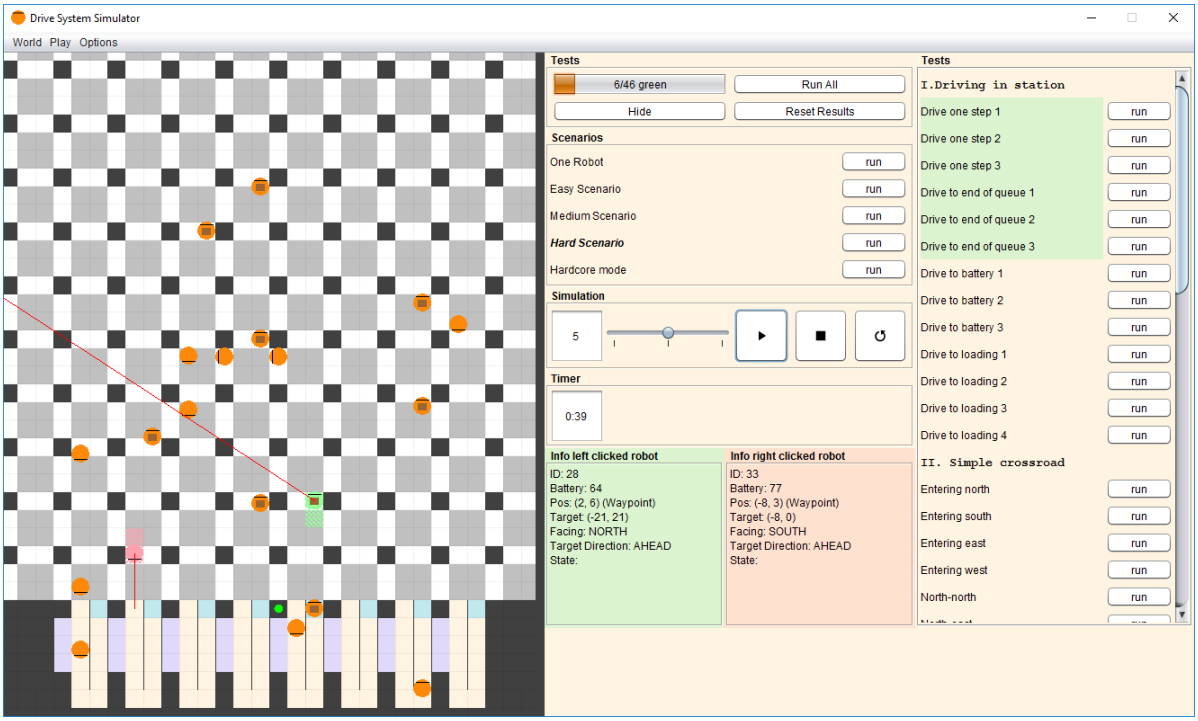


Abbildung 4: Das für das gegebene Validierungsframework.

4 Nutzung des Validierungsframeworks

Das Validierungsframework kann gestartet werden, indem im „PROJEKTNAME“-Projekt die Datei `App.java` unter `src/de.hpi.mod.sim` ausgeführt wird. Dazu muss die Datei mit einem Rechtsklick angeklickt werden und im Menü „Run as“ → „Java Application“ ausgewählt werden. Für das wiederholte Ausführen ist `App.java` dann auch in der Toolbar im „Run“-Menü hinterlegt.

4.1 Fenster

Das Framework öffnet sich in einem neuen Fenster (siehe Abbildung 4). Das Fenster ist unterteilt in die Simulationsfläche links, verschiedene Menüoptionen zum Festlegen von Einstellungen und Starten von Szenarien in der Mitte sowie eine Liste an Tests rechts.

4.2 Simulationsfläche

Die Simulationsfläche bildet eine Paketsortieranlage ab, deren Struktur im Entwurfsdokument genauer beschrieben ist. Auf dieser Fläche bewegen sich die simulierten Roboter.

Die Größe der Simulationsfläche (und damit die Anzahl der simulierten Roboter) hängt von der Fenstergröße ab. Dementsprechend kann die Fenstergröße nur verändert werden, wenn gerade keine Simulation läuft.

Zur Navigation auf der Simulationsfläche können die Pfeiltasten verwendet werden. Mit der Tastenkombination aus „Strg“ und „+“ bzw. „-“ kann hinein- bzw. hinausgezoomt werden.

4.3 Einstellungen

Im mittleren Teil des Fensters sind alle wichtigen Einstellungen zu finden.

Diese befinden sich insbesondere im Bereich „Simulation“. Dort kann per Schieberegler die Geschwindigkeit der Simulation eingestellt werden. Mit den benachbarten Schaltflächen kann das aktuell laufende Szenario bzw. Test pausiert, gestoppt oder neu gestartet werden. Zum Pausieren und Fortfahren kann auch die Leertaste verwendet werden.

Während eines laufenden Szenarios oder Tests können auf der Simulationsfläche einzelne Roboter angeklickt werden. Die Informationen des zuletzt mit links bzw. rechts angeklickten Roboters werden im unteren Bereich dargestellt.

4.4 Szenarien

Im Bereich „Scenarios“ (mitte oben) kann Simulation eines Szenarios gestartet werden. Je nach gewähltem Szenario, werden dabei ein oder mehrere Roboter auf der Simulationsfläche platziert. Diese Roboter werden durch die aktuelle Version des generierten Codes gesteuert. Es werden kontinuierlich Aufträge an die Roboter übermittelt, bis das Szenario beendet wird. Kollisionen und Deadlocks führen zum vorzeitigen Abbruch des Szenarios.

Das aktuell ausgewählte Szenario ist durch eine Hervorhebung des Namens markiert.

4.5 Tests

Ein Test prüft eine vordefinierte Fahrsituation und gibt unmittelbar eine Rückmeldung über deren Erfolg. Das Verhalten der Roboter kann dabei auf der Simulationsfläche beobachtet werden.

Für die Steuerung der Tests gibt es den Bereich „Tests“ im mittleren Teil des Fensters sowie die Liste aller Tests im rechten Teil des Fensters. Die Liste aller Tests kann mit einem Klick auf „Hide“ bzw. „Show“ aus- bzw. eingeblendet werden.

Jeder Test kann individuell über „Run“ gestartet werden. Alternativ kann per „Run All“ die gesamte Testliste automatisch abgearbeitet werden.

Die Farbe, mit der der Titel eines Tests hinterlegt wird, gibt Auskunft über den Erfolg bzw. den Misserfolg dieses Tests beim letzten versuchten Durchlauf. Einen Überblick über die Testergebnisse gibt zudem der Fortschrittsbalken im Bereich „Tests“. Die Testergebnisse bleiben auch nach dem Schließen des Simulators erhalten, damit nicht jedes Mal die gesamte Testliste neu geprüft werden muss. Die Testergebnisse können mittels „Reset Results“ gelöscht werden.