

RWTH AACHEN

CES Softwareentwicklungspraktikum

Analyse- und Entwurfsdokument - Wärmeleitung

Christian BILAS
 christian.bilas@rwth-aachen.de, Matrikelnummer: 334829

Robin Tim BROESKE
 robin.tim.broeske@rwth-aachen.de, Matrikelnummer: 334031

Konstantin KEY
 konstantin.key@rwth-aachen.de, Matrikelnummer: 332523

Inhaltsverzeichnis

Inhaltsverzeichnis	1
1 Vorwort	2
1.1 Aufgabenstellung und Struktur des Dokument	2
1.2 Projektmanagement	2
1.3 Lob und Kritik	2
2 Analyse	3
2.1 Anforderungsanalyse	3
2.1.1 Benutzeranforderungen	3
2.1.2 Anwendungsfallanalyse	3
2.1.2.1 Anwendungsfalldiagramm	3
2.1.2.2 Beschreibungen der Anwendungsfälle	4
2.1.2.3 Aktivitätsdiagramme	9
2.1.2.4 Systemanforderungen	14
2.2 Begriffsanalyse	15
2.2.1 Klassenkandidaten	15
2.2.2 Begriffsnetz	16
3 Entwurf	17
3.1 Pakete	17
3.2 Abstrakte Datentypen	17
3.3 Klassen	17
3.3.1 Paket algorithms	17
3.3.1.1 IntMethod	18
3.3.1.2 IterativeSolver	20
3.3.2 Paket model	21
3.3.2.1 model	22
3.3.3 Paket presentation	27
3.3.3.1 UI	28
3.3.3.2 Controller	32
4 Entwicklerdokumentation	41
4.1 Codestruktur	41
4.2 Detaillierte Dokumentation des Codes	41
4.3 Software Tests	41
4.3.1 algorithms	41
4.3.1.1 CRS	41
4.3.2 Solver	42
Abbildungsverzeichnis	44
Tabellenverzeichnis	45

Kapitel 1

Vorwort

1.1 Aufgabenstellung und Struktur des Dokument

1.2 Projektmanagement

1.3 Lob und Kritik

Kapitel 2

Analyse

2.1 Anforderungsanalyse

2.1.1 Benutzeranforderungen

Es soll eine Software zur Simulation der zeitlichen Entwicklung einer Temperaturverteilung in Metallplatten entwickelt werden. Diese sollen die Abmessungen 1 Meter x 1 Meter besitzen und können weiterhin bezüglich der örtlichen Verteilung der Temperaturleitkoeffizienten inhomogen sein. Des Weiteren bietet sich dem Benutzer die Möglichkeit Wärmequellen und deren Intensität einzugeben. Außerdem ist es dem Benutzer möglich zur vollständigen Spezifikation eines Simulationsexperiments die Start- und Randbedingungen des Wärmeleitungsproblems, den Endzeitpunkt der Simulation sowie die Simulationsparameter der Orts- beziehungsweise Zeitdiskretisierung vorzugeben. Jegliche Benutzereingaben erfolgen über eine grafische Oberfläche. Nach Abschluss der Berechnung wird das Ergebnis visualisiert und die zeitliche Entwicklung der Temperaturverteilung kann in Form eines Videos untersucht werden.

2.1.2 Anwendungsfallanalyse

2.1.2.1 Anwendungsfalldiagramm

Das Anwendungsfalldiagramm zeigt die Abbildung 2.1.

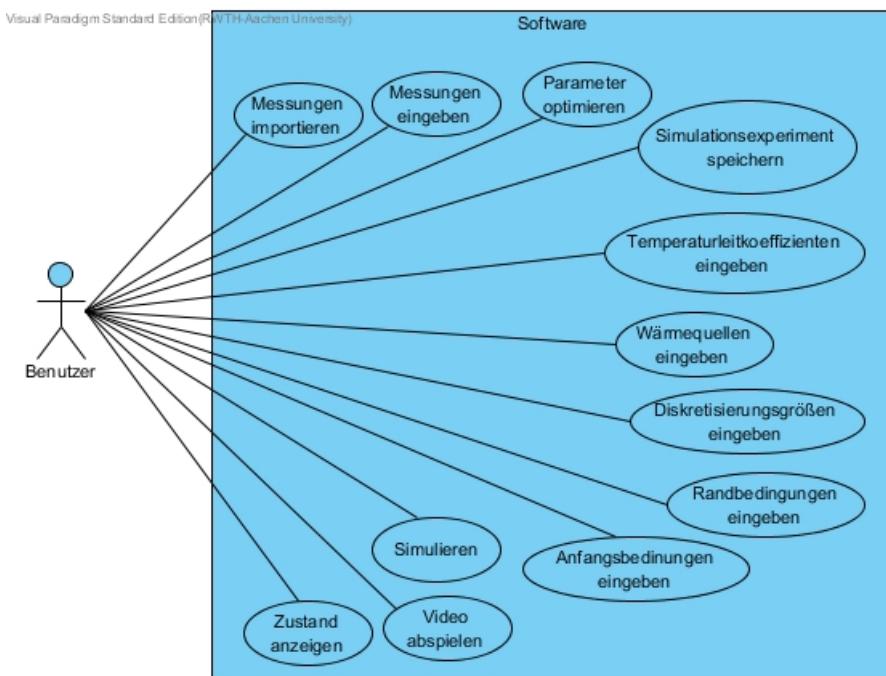


Abbildung 2.1: Anwendungsfalldiagramm

2.1.2.2 Beschreibungen der Anwendungsfälle

Die folgenden Tabellen (Tab. 2.1 - 2.12) zeigen die Beschreibungen der Anwendungsfälle.

Name	Anfangsbedingungen eingeben	
Ziel	Der Benutzer möchte Anfangsbedingungen vorgeben.	
Einordnung	Basisfunktion	
Vorbedingung	Die Software wird korrekt ausgeführt.	
Nachbedingung	Die Anfangsbedingungen wurden vorgegeben und gespeichert.	
Nachbedingung im Fehlerfall	Die Anfangsbedingungen wurden nicht geändert und entsprechende Fehlermeldungen wurden ausgegeben.	
Haupt-Neben-Akteur	Benutzer	
Auslöser	Der Benutzer möchte Anfangsbedingungen vorgeben.	
Standardfluss	Schritt	Aktion
	1	Der Benutzer wählt den Menüpunkt <i>Anfangsbedingungen</i> aus.
	2	Die Software wechselt zu dem entsprechenden Menü.
	3	Der Benutzer gibt die Anfangsbedingungen vor.
	4	Die Software prüft die eingegebenen Anfangsbedingungen.
	5	Die Software speichert die Anfangsbedingungen.
Nebenfluss	Schritt	Aktion
Anfangsbedingungen nicht akzeptiert	5a.1	Eine Fehlermeldung wird angezeigt.
	5a.2	Der Benutzer korrigiert seine Eingabe.
	5a.3	→ Schritt 4

Tabelle 2.1: Beschreibung Use Case Anfangsbedingungen eingeben

Name	Diskretisierungsgrößen eingeben	
Ziel	Der Benutzer möchte Diskretisierungsgrößen eingeben.	
Einordnung	Basisfunktion	
Vorbedingung	Die Software wird korrekt ausgeführt.	
Nachbedingung	Die Diskretisierungsgrößen wurden vorgegeben und gespeichert.	
Nachbedingung im Fehlerfall	Die Diskretisierungsgrößen wurden nicht geändert und entsprechende Fehlermeldungen wurden ausgegeben.	
Haupt-Neben-Akteur	Benutzer	
Auslöser	Der Benutzer möchte Diskretisierungsgrößen eingeben.	
Standardfluss	Schritt	Aktion
	1	Der Benutzer wählt den Menüpunkt <i>Diskretisierungsgrößen</i> aus.
	2	Die Software wechselt zu dem entsprechenden Menü.
	3	Der Benutzer gibt die Stützstellenzahl der Ortsdiskretisierung n ein.
	4	Der Benutzer gibt die Stützstellenzahl der Zeitdiskretisierung m ein.
	5	Der Benutzer gibt den Endzeitpunkt T ein.
	6	Die Software prüft die eingegebenen Größen.
	7	Die Software speichert die eingegebenen Größen.
Nebenfluss	Schritt	Aktion
Eingegebene Größen nicht akzeptiert	7a.1	Eine Fehlermeldung wird angezeigt.
	7a.2	Der Benutzer korrigiert seine Eingabe.
	7a.3	→ Schritt 6

Tabelle 2.2: Beschreibung Use Case Diskretisierungsgrößen eingeben

Name	Messungen eingeben	
Ziel	Der Benutzer möchte experimentelle Daten eingeben.	
Einordnung	Basisfunktion	
Vorbedingung	Die Software wird korrekt ausgeführt.	
Nachbedingung	Die eingegebenen Daten wurden gespeichert.	
Nachbedingung im Fehlerfall	Die Daten wurden nicht geändert und entsprechende Fehlermeldungen wurden ausgegeben.	
Haupt-Neben-Akteur	Benutzer	
Auslöser	Der Benutzer möchte Messdaten eingeben.	
Standardfluss	Schritt	Aktion
	1	Der Benutzer wählt den Menüpunkt <i>Optimierung</i> aus.
	2	Die Software wechselt zu dem entsprechenden Menü.
	3	Der Benutzer gibt die Messungen ein.
	4	Die Software prüft die eingegebenen Messwerte.
	5	Die Software speichert die Daten.
Nebenfluss	Schritt	Aktion
Randbedingungen nicht akzeptiert	5a.1	Eine Fehlermeldung wird angezeigt.
	5a.2	Der Benutzer korrigiert seine Eingabe.
	5a.3	→ Schritt 4

Tabelle 2.3: Beschreibung Use Case Messungen eingeben

Name	Messungen importieren	
Ziel	Der Benutzer möchte experimentelle Daten importieren.	
Einordnung	Basisfunktion	
Vorbedingung	Die Software wird korrekt ausgeführt.	
Nachbedingung	Die ausgewählten Daten wurden geladen.	
Nachbedingung im Fehlerfall	Die Daten wurden nicht geändert und entsprechende Fehlermeldungen wurden ausgegeben.	
Haupt-Neben-Akteur	Benutzer	
Auslöser	Der Benutzer möchte Messdaten importieren.	
Standardfluss	Schritt	Aktion
	1	Der Benutzer wählt den Menüpunkt <i>Optimierung</i> aus.
	2	Die Software wechselt zu dem entsprechenden Menü.
	3	Der Benutzer wählt die entsprechende Datei aus.
	4	Die Software prüft die angegebene Datei.
	5	Die Software wertet die Datei aus.
Nebenfluss	Schritt	Aktion
Randbedingungen nicht akzeptiert	5a.1	Eine Fehlermeldung wird angezeigt.
	5a.2	Der Benutzer korrigiert seine Eingabe.
	5a.3	→ Schritt 4

Tabelle 2.4: Beschreibung Use Case Messungen importieren

Name	Randbedingungen eingeben	
Ziel	Der Benutzer möchte Randbedingungen vorgeben.	
Einordnung	Basisfunktion	
Vorbedingung	Die Software wird korrekt ausgeführt.	
Nachbedingung	Die Randbedingungen wurden vorgegeben und gespeichert.	
Nachbedingung im Fehlerfall	Die Randbedingungen wurden nicht geändert und entsprechende Fehlermeldungen wurden ausgegeben.	
Haupt-Neben-Akteur	Benutzer	
Auslöser	Der Benutzer möchte Randbedingungen vorgeben.	
Standardfluss	Schritt	Aktion
	1	Der Benutzer wählt den Menüpunkt <i>Randbedingungen</i> aus.
	2	Die Software wechselt zu dem entsprechenden Menü.
	3	Der Benutzer gibt die Randbedingungen vor.
	4	Die Software prüft die eingegebenen Randbedingungen.
	5	Die Software speichert die Randbedingungen.
Nebenfluss	Schritt	Aktion
Randbedingungen nicht akzeptiert	5a.1	Eine Fehlermeldung wird angezeigt.
	5a.2	Der Benutzer korrigiert seine Eingabe.
	5a.3	→ Schritt 4

Tabelle 2.5: Beschreibung Use Case Randbedingungen eingeben

Name	Parameter optimieren	
Ziel	Der Benutzer möchte die Temperaturleitkoeffizienten optimal im Sinne kleinsten Fehlerquadrate an experimentelle Daten anpassen lassen.	
Einordnung	Hauptfunktion	
Vorbedingung	Die Software wird korrekt ausgeführt.	
Nachbedingung	Die Parameter wurden optimiert.	
Nachbedingung im Fehlerfall	Die Daten wurden nicht geändert und entsprechende Fehlermeldungen wurden ausgegeben.	
Haupt-Neben-Akteur	Benutzer	
Auslöser	Der Benutzer möchte die Temperaturleitkoeffizienten optimieren lassen.	
Standardfluss	Schritt	Aktion
	1	Der Benutzer wählt den Menüpunkt <i>Optimierung</i> aus.
	2	Die Software wechselt zu dem entsprechenden Menü.
	3	Der Benutzer startet die Optimierung.
	4	Die Software setzt das Ergebnis als aktuelle Temperaturleitkoeffizienten.

Tabelle 2.6: Beschreibung Use Case Parameter optimieren

Name	Simulationsexperiment laden/speichern	
Ziel	Der Benutzer möchte ein Simulationsexperiment laden/spichern.	
Einordnung	Hauptfunktion	
Vorbedingung	Die Software wird korrekt ausgeführt.	
Nachbedingung	Das eingegebene Simulationsexperiment wurde geladen/gespeichert.	
Nachbedingung im Fehlerfall	Die Daten wurden nicht geändert und entsprechende Fehlermeldungen wurden ausgegeben.	
Haupt-Neben-Akteur	Benutzer	
Auslöser	Der Benutzer möchte ein Simulationsexperiment laden/speichern.	
Standardfluss	Schritt	Aktion
	1	Der Benutzer wählt den Menüpunkt <i>Simulation</i> aus.
	2	Die Software wechselt zu dem entsprechenden Menü.
	3	Der Benutzer wählt die gewünschte Datei aus.
	4	Die Software prüft die angegebene Datei.
	5	Die Software lädt/speichert das Simulationsexperiment.
Nebenfluss	Schritt	Aktion
Randbedingungen nicht akzeptiert	5a.1	Eine Fehlermeldung wird angezeigt.
	5a.2	Der Benutzer korrigiert seine Eingabe.
	5a.3	→ Schritt 4

Tabelle 2.7: Beschreibung Use Case Simulationsexperiment laden/speichern

Name	Simulieren	
Ziel	Der Benutzer möchte simulieren.	
Einordnung	Hauptfunktion	
Vorbedingung	Die Software wird korrekt ausgeführt.	
Nachbedingung	Die Simulation wurde ausgeführt.	
Nachbedingung im Fehlerfall	Die Simulation wurden nicht ausgeführt und entsprechende Fehlermeldungen wurden ausgegeben.	
Haupt-Neben-Akteur	Benutzer	
Auslöser	Der Benutzer möchte die Simulation starten.	
Standardfluss	Schritt	Aktion
	1	Der Benutzer wählt den Menüpunkt <i>Simulieren</i> aus.
	2	Die Software wechselt zu dem entsprechenden Menü.
	3	Der Benutzer drückt den Knopf <i>Simulieren</i> .
	4	Die Software simuliert.
	5	Die Software wechselt zu dem Menü <i>Visualisierung</i> .
	6	Die Software stellt den Endzustand dar.

Tabelle 2.8: Beschreibung Use Case Simulieren

Name	Temperaturleitkoeffizienten eingeben	
Ziel	Der Benutzer möchte Temperaturleitkoeffizienten eingeben.	
Einordnung	Basisfunktion	
Vorbedingung	Die Software wird korrekt ausgeführt.	
Nachbedingung	Die Temperaturleitkoeffizienten wurden eingegeben und gespeichert.	
Nachbedingung im Fehlerfall	Die Temperaturleitkoeffizienten wurden nicht geändert und entsprechende Fehlermeldungen wurden ausgegeben.	
Haupt-Neben-Akteur	Benutzer	
Auslöser	Der Benutzer möchte Temperaturleitkoeffizienten eingeben.	
Standardfluss	Schritt	Aktion
	1	Der Benutzer wählt den Menüpunkt <i>Temperaturleitkoeffizienten</i> aus.
	2	Die Software wechselt zu dem entsprechenden Menü.
	3	Der Benutzer wählt auf der Darstellung der Platte die gewünschten Gebiete.
	4	Die Software prüft die eingegebenen Gebiete.
	5	Der Benutzer wählt die Werte für die einzelnen Gebiete.
	6	Die Software prüft die eingegebenen Werte.
	7	Die Software speichert die Gebiete und die Werte.
Nebenfluss	Schritt	Aktion
Gebiet nicht akzeptiert	5a.1	Eine Fehlermeldung wird angezeigt.
	5a.2	Der Benutzer korrigiert seine Eingabe.
	5a.3	→ Schritt 4
Werte nicht akzeptiert	7a.1	Eine Fehlermeldung wird angezeigt.
	7a.2	Der Benutzer korrigiert seine Eingabe.
	7a.3	→ Schritt 6

Tabelle 2.9: Beschreibung Use Case Temperaturleitkoeffizienten eingeben

Name	Video abspielen	
Ziel	Der Benutzer möchte die zeitliche Entwicklung der Temperaturverteilung untersuchen.	
Einordnung	Hauptfunktion	
Vorbedingung	Die Software wird korrekt ausgeführt und es wurde eine Simulation erfolgreich durchgeführt.	
Nachbedingung	Das Video wird abgespielt.	
Nachbedingung im Fehlerfall	Das Video wurde nicht abgespielt und entsprechende Fehlermeldungen wurden ausgegeben.	
Haupt-Neben-Akteur	Benutzer	
Auslöser	Der Benutzer möchte die zeitliche Entwicklung der Temperaturverteilung untersuchen.	
Standardfluss	Schritt	Aktion
	1	Der Benutzer wählt den Menüpunkt <i>Visualisierung</i> aus.
	2	Die Software wechselt zu dem entsprechenden Menü.
	3	Der Benutzer startet das Video.
	4	Die Software spielt das Video ab.

Tabelle 2.10: Beschreibung Use Case Video abspielen

Name	Wärmequellen eingeben	
Ziel	Der Benutzer möchte Wärmequellen eingeben.	
Einordnung	Basisfunktion	
Vorbedingung	Die Software wird korrekt ausgeführt.	
Nachbedingung	Die Wärmequellen wurden eingegeben und gespeichert.	
Nachbedingung im Fehlerfall	Die Wärmequellen wurden nicht geändert und entsprechende Fehlermeldungen wurden ausgegeben.	
Haupt-Neben-Akteur	Benutzer	
Auslöser	Der Benutzer möchte Wärmequellen eingeben.	
Standardfluss	Schritt	Aktion
	1	Der Benutzer wählt den Menüpunkt <i>Wärmequellen</i> aus.
	2	Die Software wechselt zu dem entsprechenden Menü.
	3	Der Benutzer wählt auf der Darstellung der Platte die gewünschten Gebiete.
	4	Die Software prüft die eingegebenen Gebiete.
	5	Der Benutzer wählt die Werte für die einzelnen Gebiete.
	6	Die Software prüft die eingegebenen Werte.
	7	Die Software speichert die Gebiete sowie die Werte.
Nebenfluss	Schritt	Aktion
Gebiet nicht akzeptiert	5a.1	Eine Fehlermeldung wird angezeigt.
	5a.2	Der Benutzer korrigiert seine Eingabe.
	5a.3	→ Schritt 4
Werte nicht akzeptiert	7a.1	Eine Fehlermeldung wird angezeigt.
	7a.2	Der Benutzer korrigiert seine Eingabe.
	7a.3	→ Schritt 6

Tabelle 2.11: Beschreibung Use Case Wärmequellen eingeben

Name	Zustand anzeigen	
Ziel	Der Benutzer möchte ein Zustand anzeigen lassen.	
Einordnung	Hauptfunktion	
Vorbedingung	Die Software wird korrekt ausgeführt und es wurde eine Simulation erfolgreich durchgeführt.	
Nachbedingung	Der Zustand wird angezeigt.	
Nachbedingung im Fehlerfall	Der Zustand wurde nicht angezeigt und entsprechende Fehlermeldungen wurden ausgegeben.	
Haupt-Neben-Akteur	Benutzer	
Auslöser	Der Benutzer möchte ein Zustand anzeigen lassen.	
Standardfluss	Schritt	Aktion
	1	Der Benutzer wählt den Menüpunkt <i>Visualisierung</i> aus.
	2	Die Software wechselt zu dem entsprechenden Menü.
	3	Der Benutzer wählt per Maus den Zeitpunkt des Zustands, den er betrachten möchte, aus.
	4	Die Software zeigt den Zustand an.

Tabelle 2.12: Beschreibung Use Case Zustand anzeigen

2.1.2.3 Aktivitätsdiagramme

Die folgenden Abbildungen (Abb. 2.2 - 2.13) zeigen die Aktivitätsdiagramme der Anwendungsfälle.

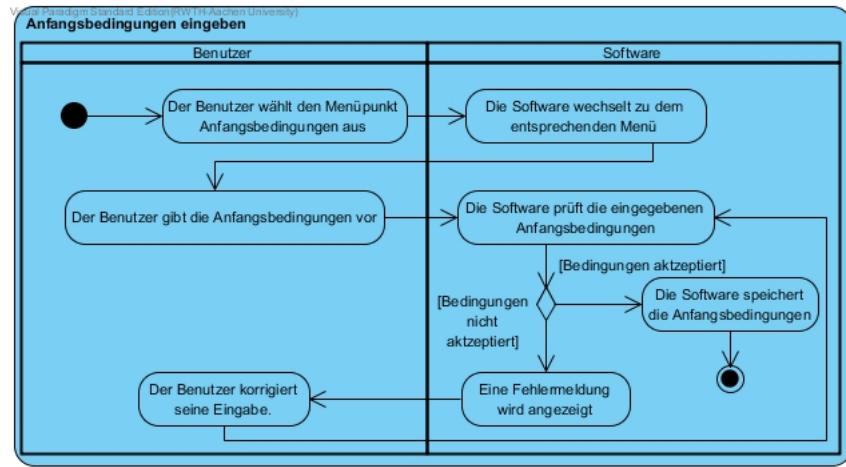


Abbildung 2.2: Aktivitätsdiagramm Use Case Anfangsbedingungen eingeben

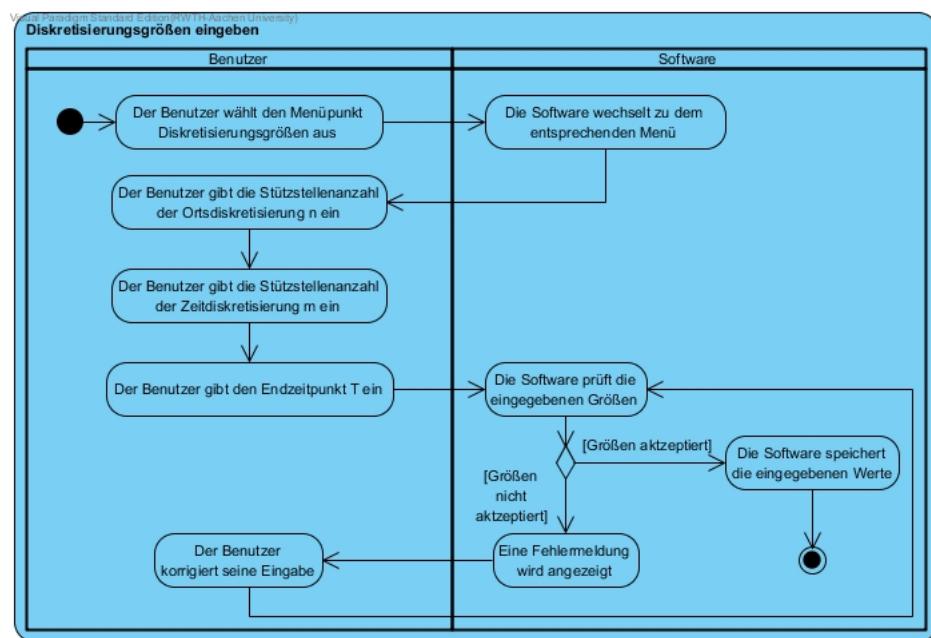


Abbildung 2.3: Aktivitätsdiagramm Use Case Diskretisierungsgrößen eingeben

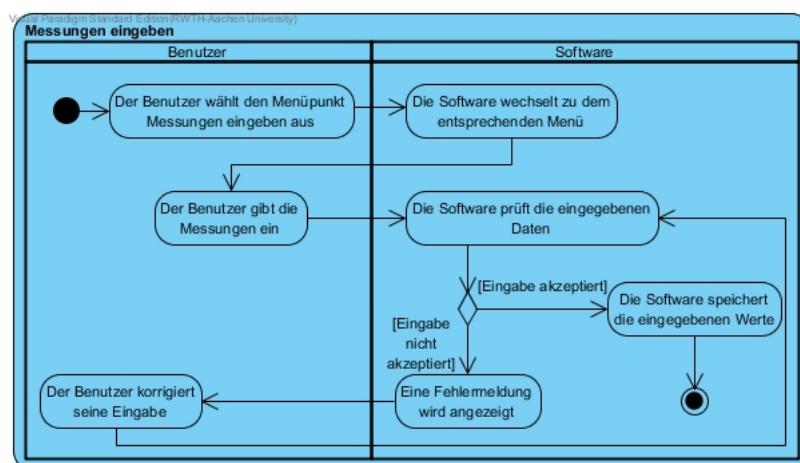


Abbildung 2.4: Aktivitätsdiagramm Use Case Messungen eingeben

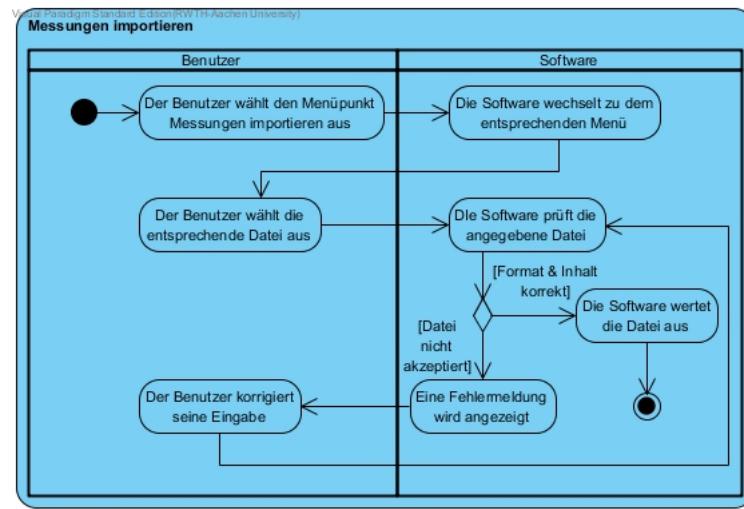


Abbildung 2.5: Aktivitätsdiagramm Use Case Messungen importieren

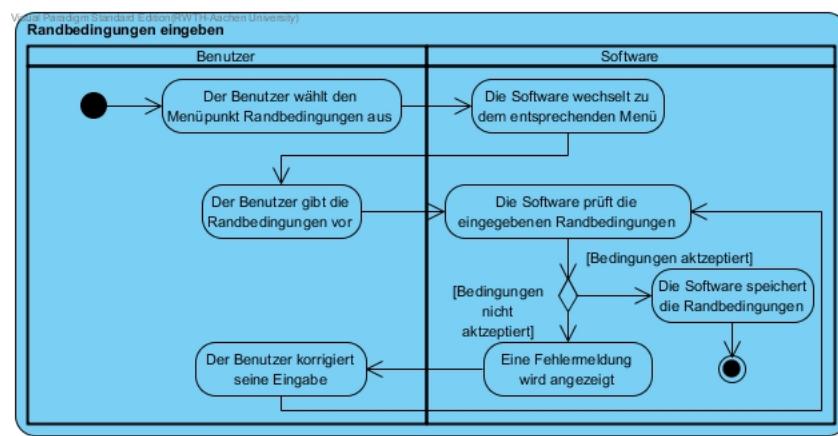


Abbildung 2.6: Aktivitätsdiagramm Use Case Randbedingungen eingeben

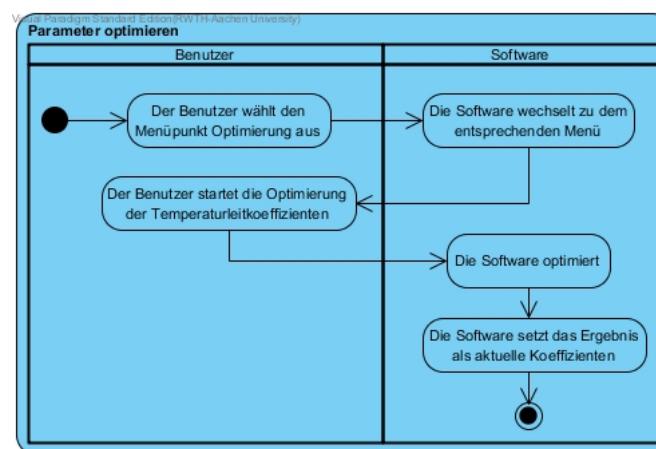


Abbildung 2.7: Aktivitätsdiagramm Use Case Parameter optimieren

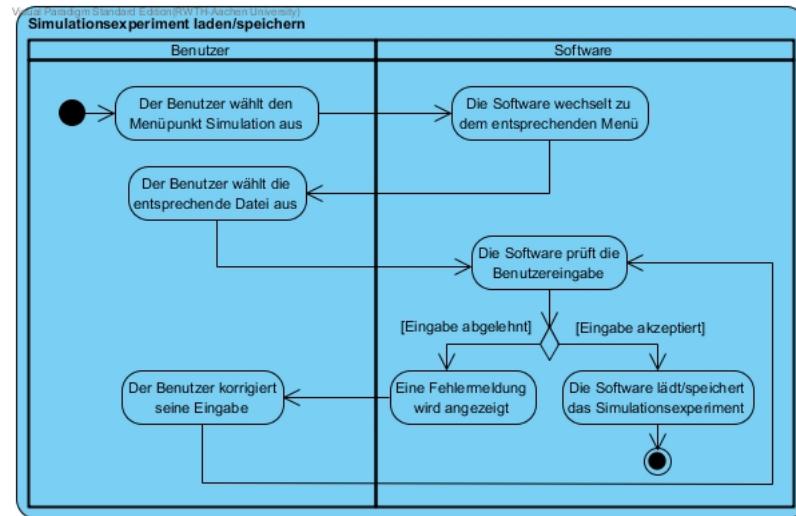


Abbildung 2.8: Aktivitätsdiagramm Use Case Simulationsexperiment laden/speichern

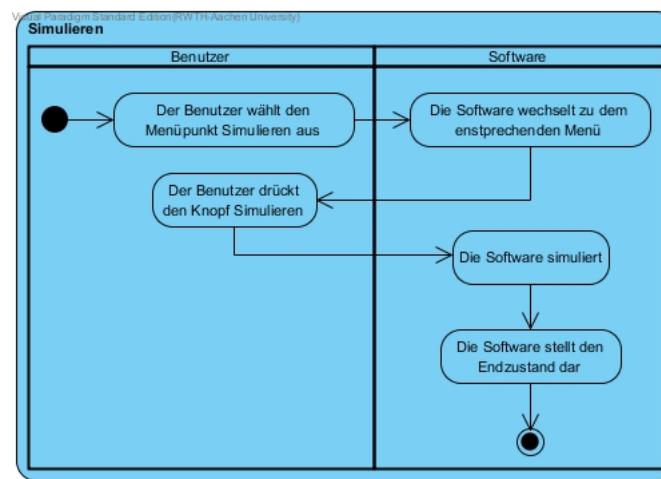


Abbildung 2.9: Aktivitätsdiagramm Use Case Simulieren

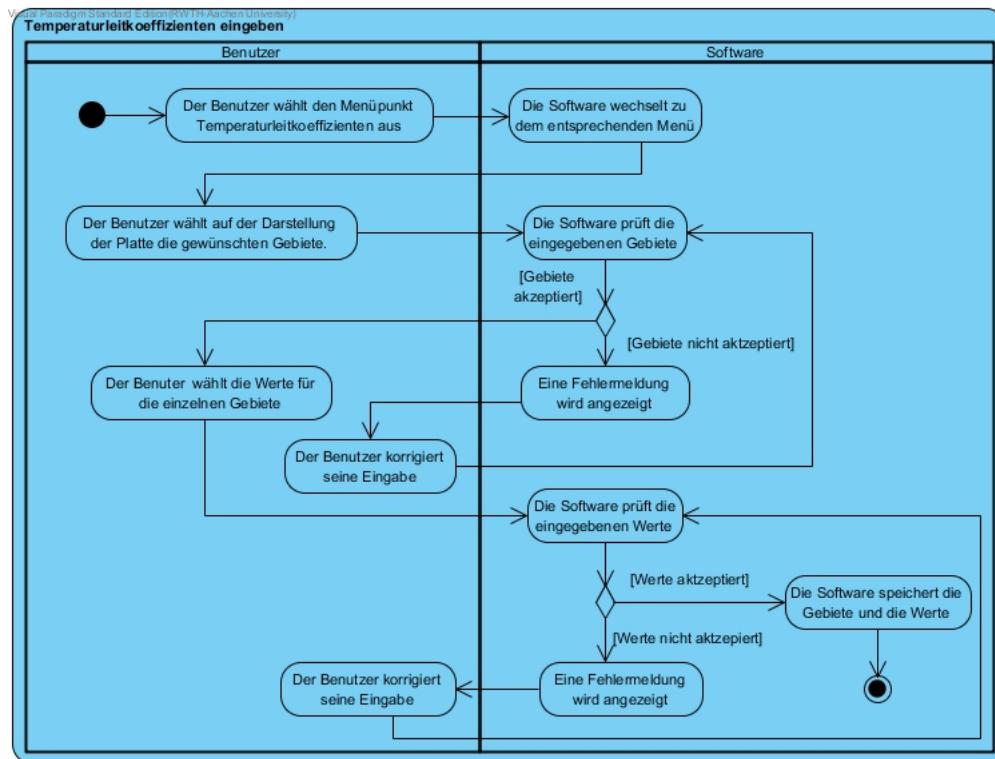


Abbildung 2.10: Aktivitätsdiagramm Use Case Temperaturleitkoeffizienten eingeben

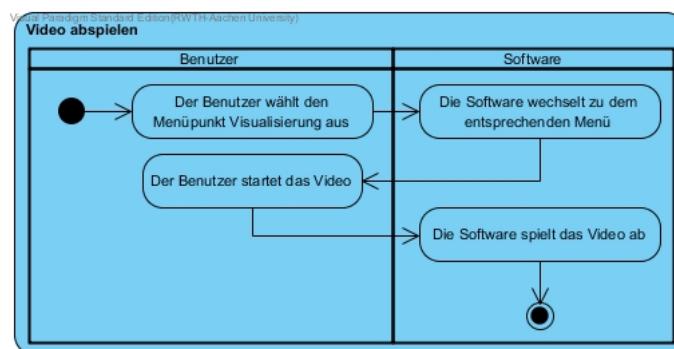


Abbildung 2.11: Aktivitätsdiagramm Use Case Video abspielen

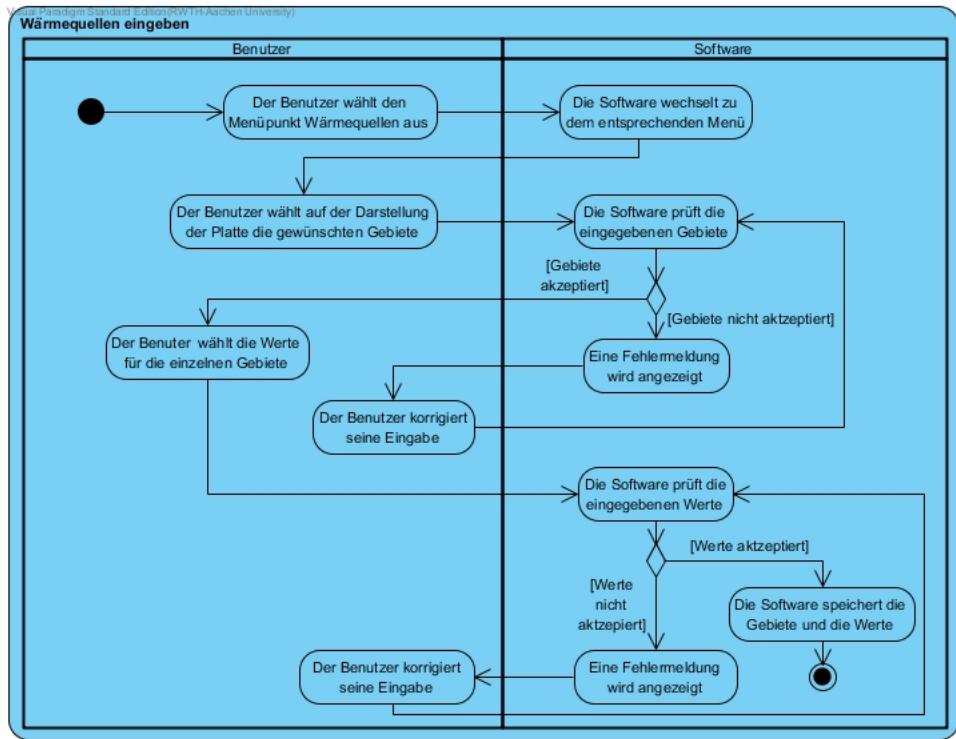


Abbildung 2.12: Aktivitätsdiagramm Use Case Wärmequellen eingeben

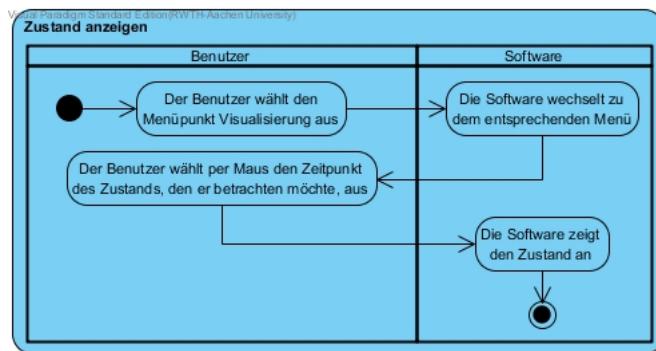


Abbildung 2.13: Aktivitätsdiagramm Use Case Zustand anzeigen

2.1.2.4 Systemanforderungen

Funktionale Anforderungen

- Der Benutzer kann mit linken Mausklicks Gebiete der Temperaturleitkoeffizienten eingeben und deren Werte per Tastatur festlegen.
- Der Benutzer kann mit linken Mausklicks Wärmequellen eingeben und deren Werte per Tastatur festlegen.
- Um das Problem zu spezifizieren, kann der Benutzer die Anfangs- und Randbedingungen vorgeben.
- Die Diskretisierungsparameter (Stützstellenzahlen der Orts- beziehungsweise Zeitdiskretisierungen sowie den Endzeitpunkt der Simulation) & Simulationsparameter (Integrationsverfahren, LGS-Löser) können durch den Benutzer festgelegt werden.
- Die Simulation kann per Knopfdruck durch den Benutzer gestartet werden.

6. Der Benutzer kann sich die zeitliche Entwicklung der Temperaturverteilung als Video oder einen Zustand als Standbild anzeigen lassen.
7. Der Benutzer kann sich eine Hilfe zur Benutzung der Software anzeigen lassen.

Nicht-funktionale Anforderungen

1. Dokumentation der Implementierung mittels Doxygen
2. Grafische Oberfläche mit Qt
3. Einfache Erweiterbarkeit um weitere Simulationsmethoden
4. Lauffähig unter Windows und Linux (insbesondere auf dem RWTH Aachen Cluster)
5. Grafische Oberfläche skaliert korrekt bei Veränderung der Fenstergröße

2.2 Begriffsanalyse

2.2.1 Klassenkandidaten

- Platte → Gitter
- Temperaturverteilung
- Temperaturleitkoeffizient (\rightarrow durch *Area* implementiert)
- Wärmequellen (\rightarrow durch *Area* implementiert)
- Startbedingung
- Randbedingung
- Endzeitpunkt, Stützstellenzahl (Ort- & Zeitdiskretisierung)
- Simulation
- Problem + Ergebnis → **Model**
- Zustand/Video
- Fehlermeldung (\rightarrow durch GUI implementiert)
- **Area**
- **IntMethod** → **ImpEuler**, ...
- **Solver** → **Jacobi**, ...

2.2.2 Begriffsnetz

Abbildung 2.14 zeigt das Begriffsnetz.

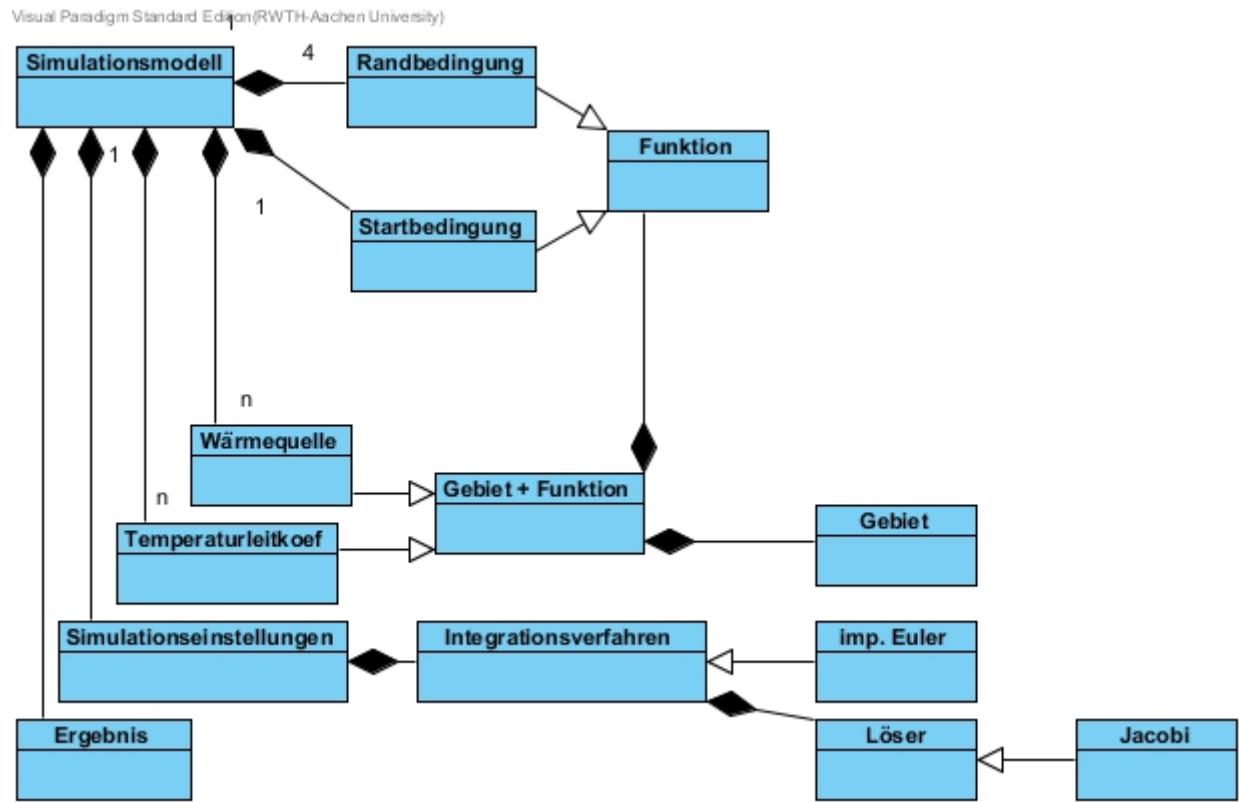


Abbildung 2.14: Begriffsnetz

Kapitel 3

Entwurf

3.1 Pakete

Unsere Software gliedert sich in drei Pakete, deren Struktur in Abbildung 3.1 dargestellt ist.

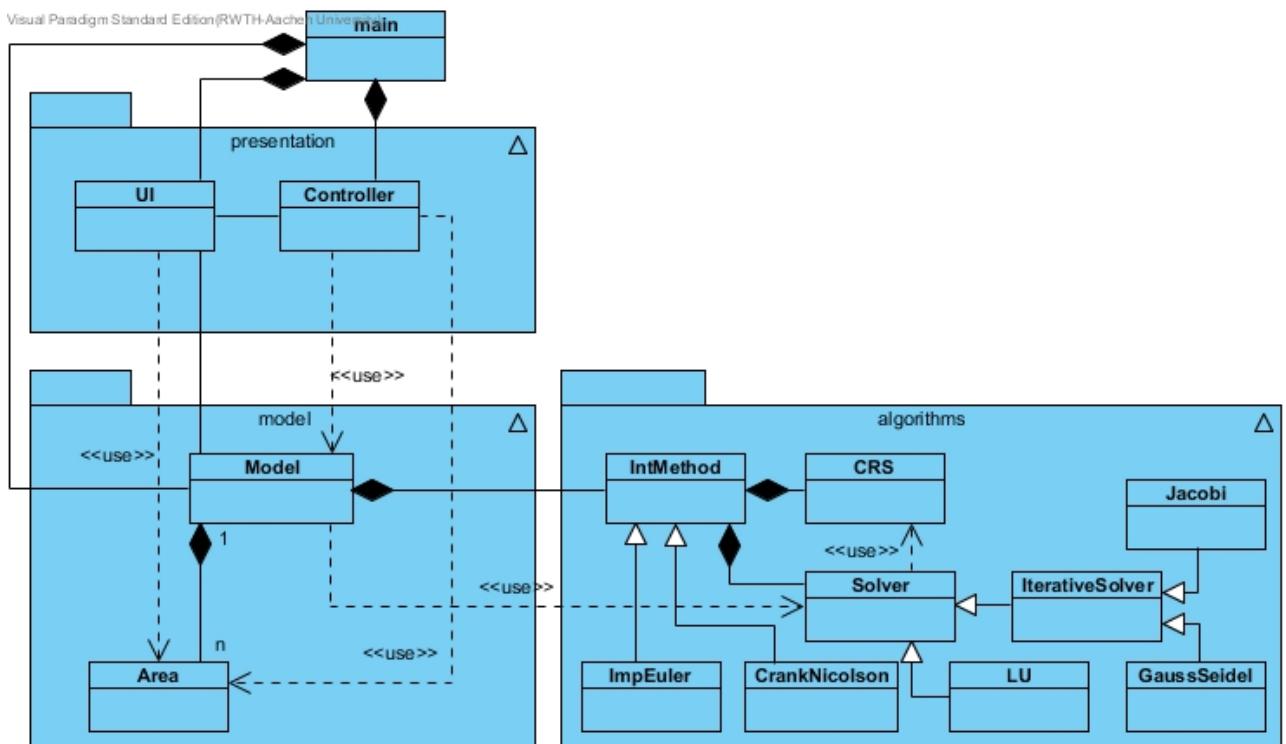


Abbildung 3.1: Paketstruktur

3.2 Abstrakte Datentypen

3.3 Klassen

Nachfolgend sind die Klassen-/Sequenzdiagramme nach Paketen sortiert aufgelistet.

Dabei werden keine Sequenzdiagramme gezeigt, falls es sich um Methoden ohne Kommunikation mit anderen Objekten handelt, insbesondere auch getter-Funktionen oder Funktionen die Algorithmen implementieren.

3.3.1 Paket algorithms

Das Klassendiagramm in Abbildung 3.2 zeigt alle im Paket *algorithms* enthaltene Klassen.

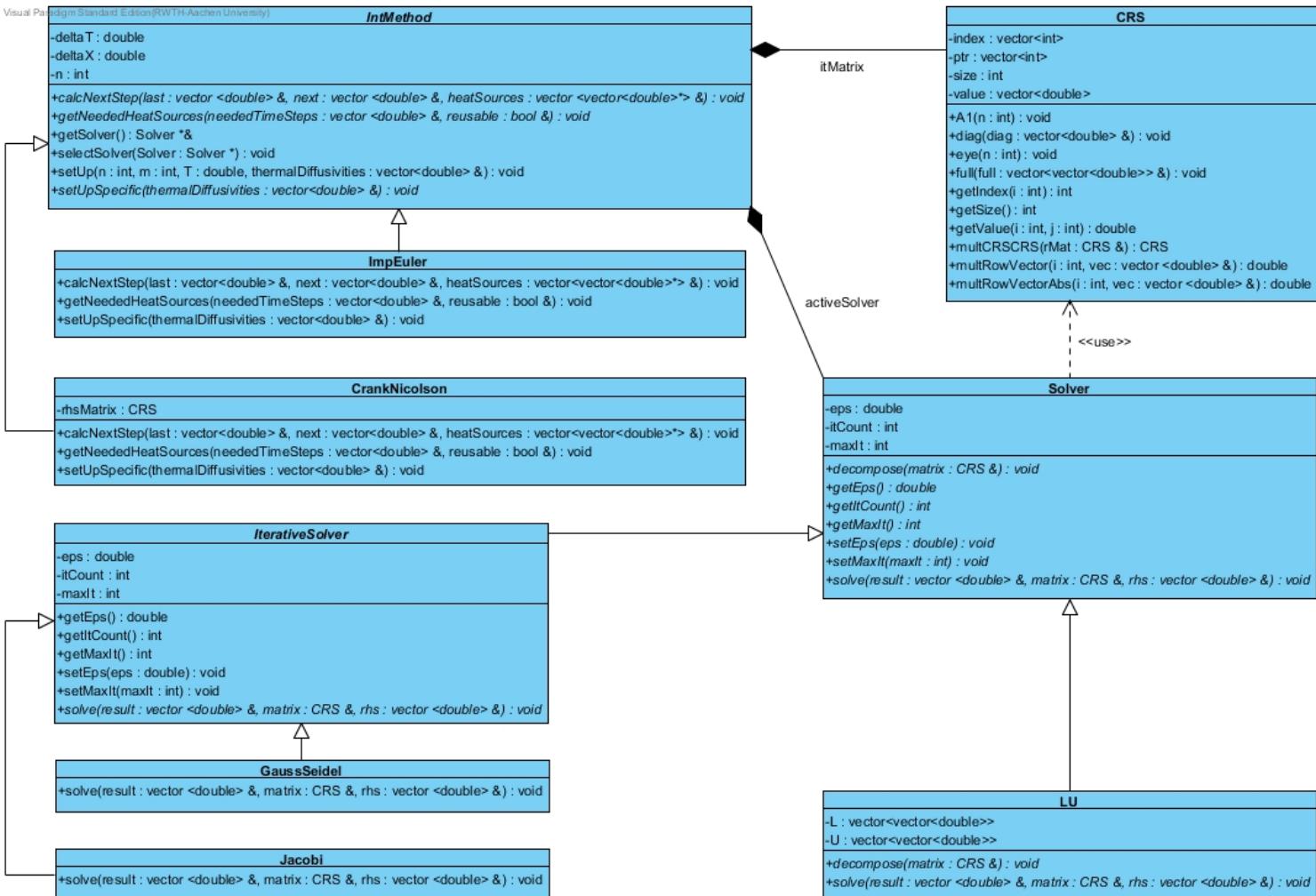


Abbildung 3.2: Klassendiagramm algorithms

3.3.1.1 IntMethod

calcNextStep

Das Sequenzdiagramm für *calcNextStep* ist in Abbildung 3.3 dargestellt. *calcNextStep* berechnet die Approximation der Temperaturverteilung zum nächsten Zeitpunkt unter Verwendung der aktuellen Verteilung sowie der eingegebenen Temperaturleitkoeffizienten und Wärmequellen.

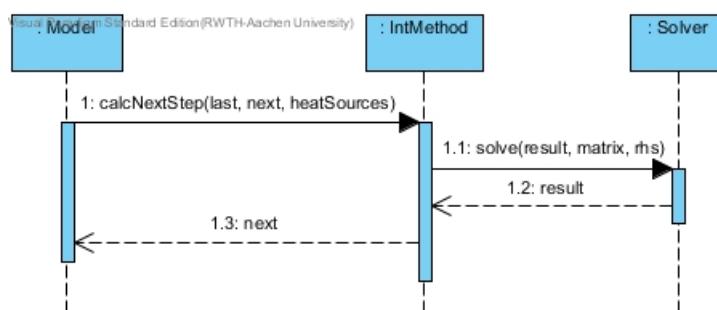


Abbildung 3.3: Sequenzdiagramm calcNextStep

selectSolver

Das Sequenzdiagramm für *selectSolver* ist in Abbildung 3.4 dargestellt. *selectSolver* setzt den gewählten Löser für lineare Gleichungssysteme.

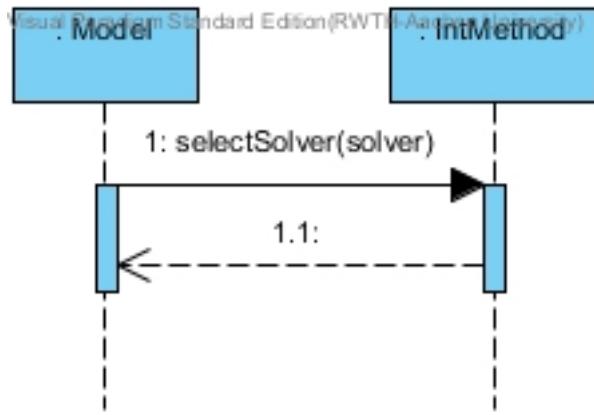


Abbildung 3.4: Sequenzdiagramm `selectSolver`

`setUp`

Das Sequenzdiagramm für `setUp` ist in Abbildung 3.5 dargestellt. `setUp` bereitet die Simulationsberechnung vor.

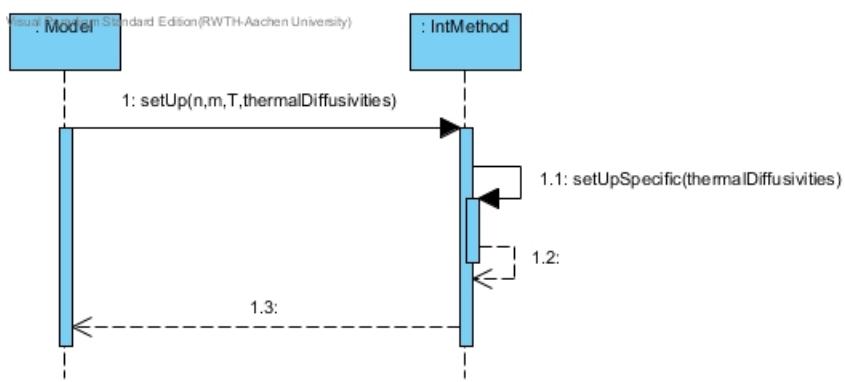


Abbildung 3.5: Sequenzdiagramm `setUp`

`setUpSpecific`

Das Sequenzdiagramm für `setUpSpecific` ist in Abbildung 3.6 dargestellt. `setUpSpecific` trifft für die gewählte Integrationsmethode spezifische Vorbereitungen.

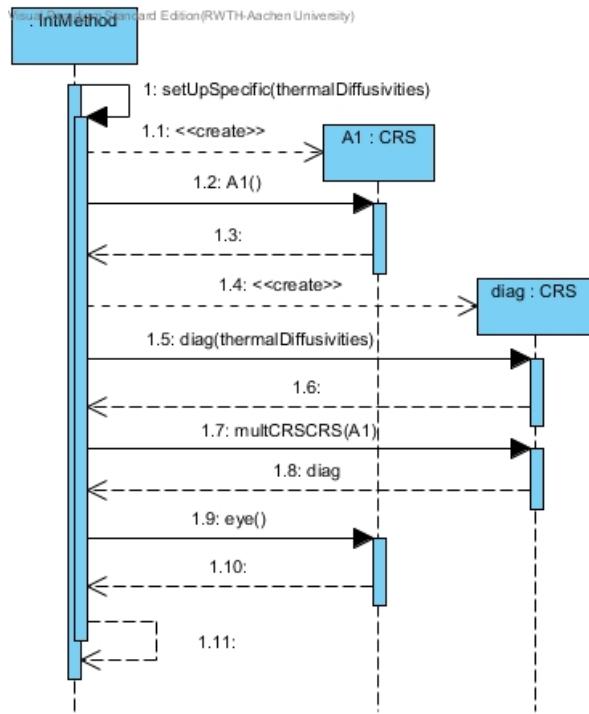


Abbildung 3.6: Sequenzdiagramm `setUpSpecific`

3.3.1.2 IterativeSolver

setEps

Das Sequenzdiagramm für `setEps` ist in Abbildung 3.7 dargestellt. `setEps` setzt die gewählte relative Genauigkeit.

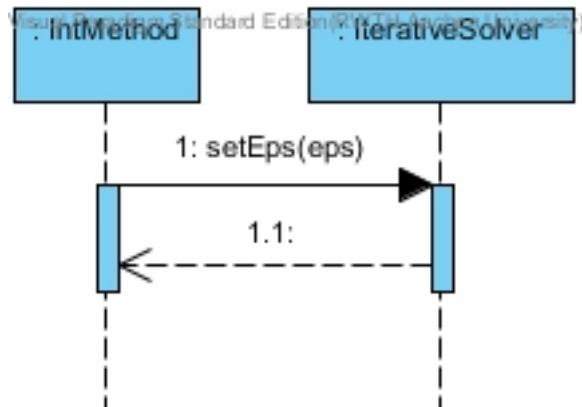


Abbildung 3.7: Sequenzdiagramm `setEps`

setMaxIt

Das Sequenzdiagramm für `setMaxIt` ist in 3.8 dargestellt. `setMaxIt` setzt die gewählte maximale Iterationsanzahl.

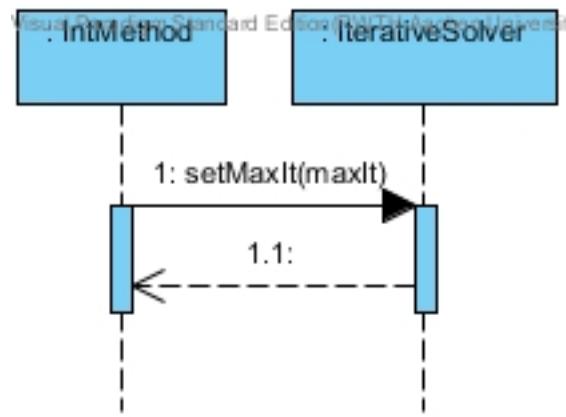


Abbildung 3.8: Sequenzdiagramm `setMaxIt`

3.3.2 Paket model

Das Klassendiagramm in Abbildung 3.9 zeigt alle im Paket *model* enthaltene Klassen.

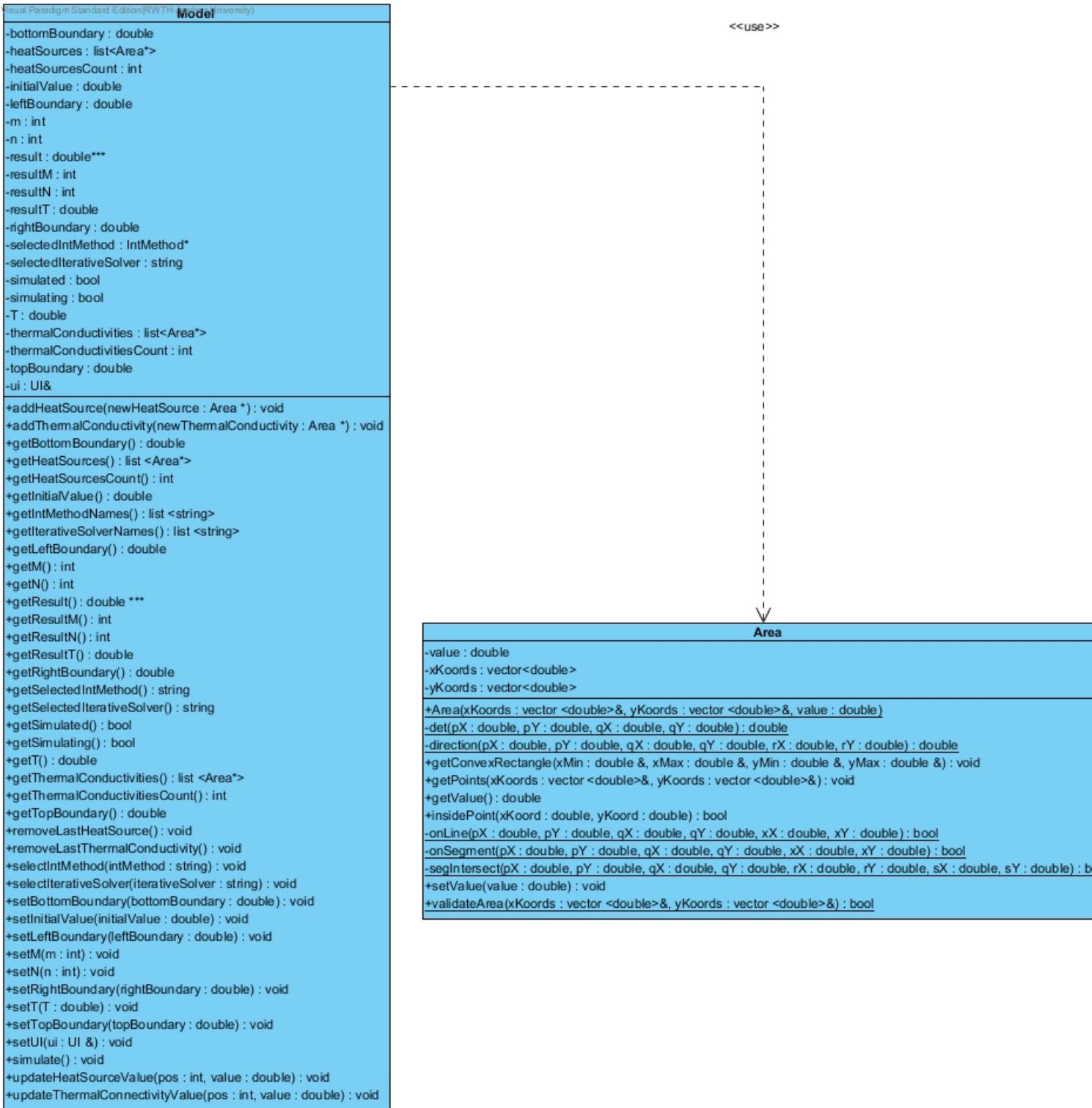


Abbildung 3.9: Klassendiagramm model

3.3.2.1 model

addHeatSource

Das Sequenzdiagramm für *addHeatSource* ist in 3.10 dargestellt. *addHeatSource* fügt eine weitere Wärmequelle hinzu.

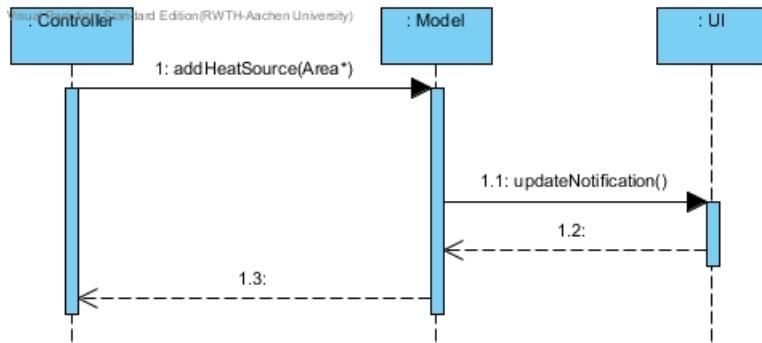


Abbildung 3.10: Sequenzdiagramm addHeatSource

addThermalDiffusivity

Das Sequenzdiagramm für `addThermalDiffusivity` ist in 3.11 dargestellt. `addThermalDiffusivity` fügt ein durch den Nutzer gewähltes Gebiet mit zugehörigem Temperaturleitkoeffizienten hinzu.

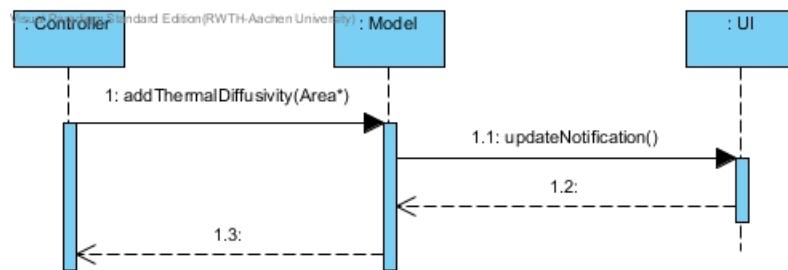


Abbildung 3.11: Sequenzdiagramm addThermalDiffusivity

removeLastHeatSource

Das Sequenzdiagramm für `removeLastHeatSource` ist in 3.12 dargestellt.

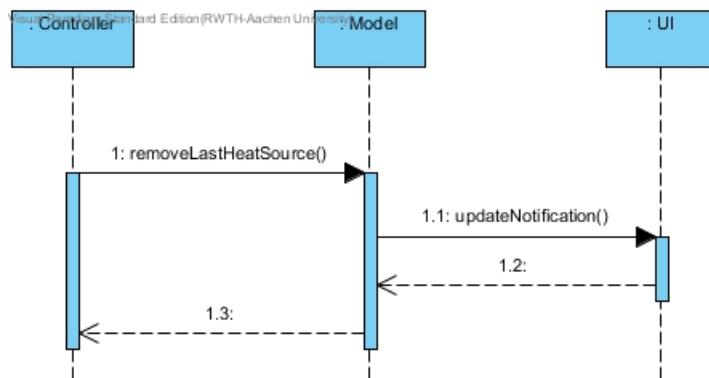


Abbildung 3.12: Sequenzdiagramm removeLastHeatSource

removeLastThermalDiffusivity

Das Sequenzdiagramm für `removeLastThermalDiffusivity` ist in 3.13 dargestellt.

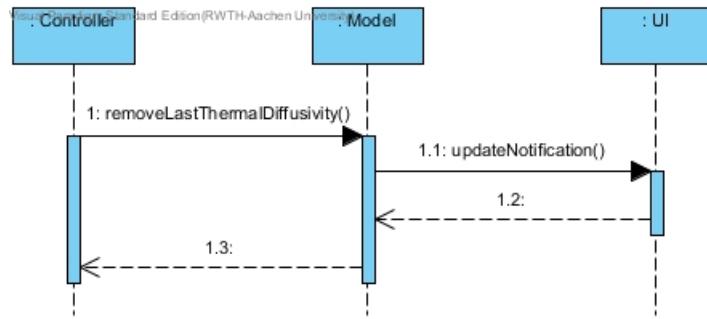


Abbildung 3.13: Sequenzdiagramm `removeLastThermalDiffusivity`

selectIntMethod

Das Sequenzdiagramm für `selectIntMethod` ist in 3.14 dargestellt.

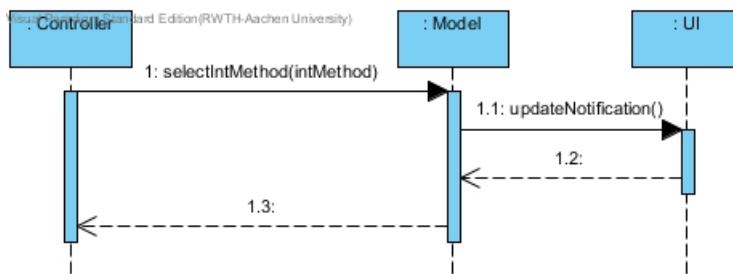


Abbildung 3.14: Sequenzdiagramm `selectIntMethod`

selectSolver

Das Sequenzdiagramm für `selectSolver` ist in 3.15 dargestellt.

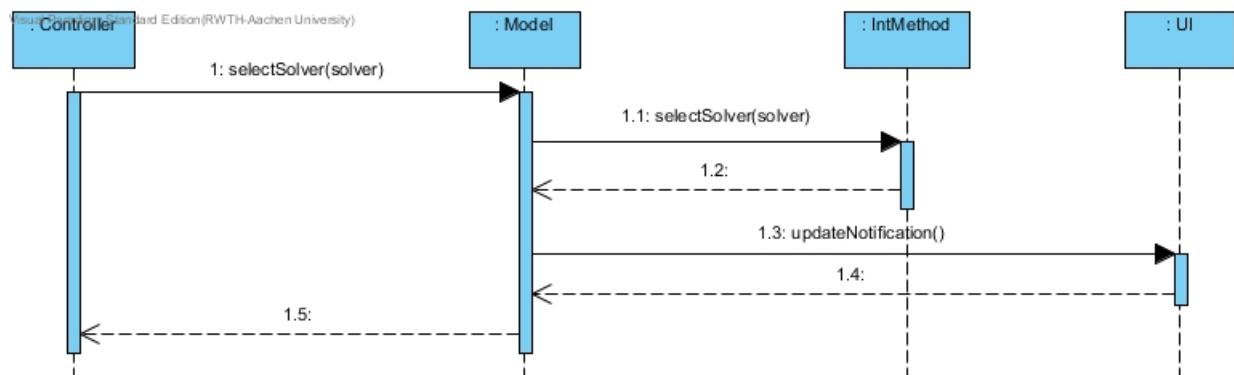


Abbildung 3.15: Sequenzdiagramm `selectSolver`

setBottomBoundary

Das Sequenzdiagramm für `setBottomBoundary` ist in 3.16 dargestellt.

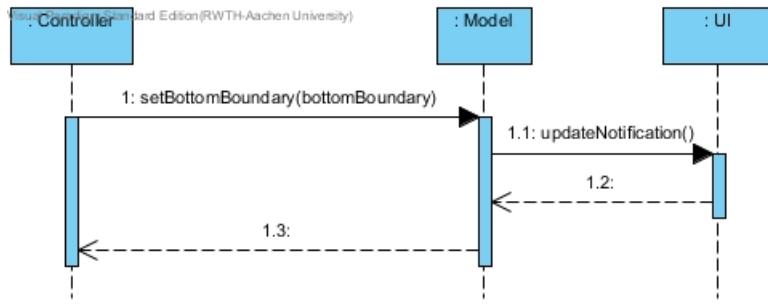


Abbildung 3.16: Sequenzdiagramm `setBottomBoundary`

`setInitialValue`

Das Sequenzdiagramm für `setInitialValue` ist in 3.17 dargestellt.

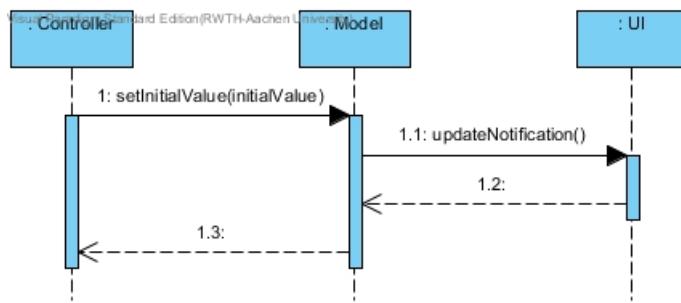


Abbildung 3.17: Sequenzdiagramm `setInitialValue`

`setLeftBoundary`

Das Sequenzdiagramm für `setLeftBoundary` ist in 3.18 dargestellt.

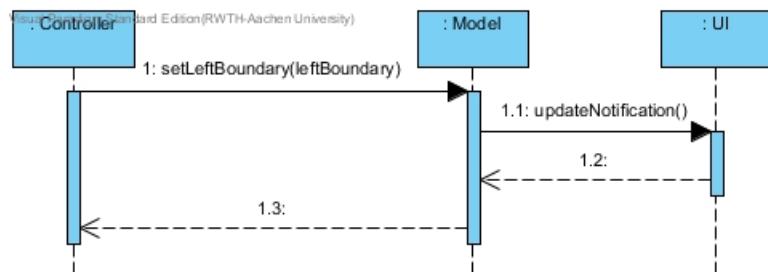


Abbildung 3.18: Sequenzdiagramm `setLeftBoundary`

`setM`

Das Sequenzdiagramm für `setM` ist in 3.19 dargestellt.

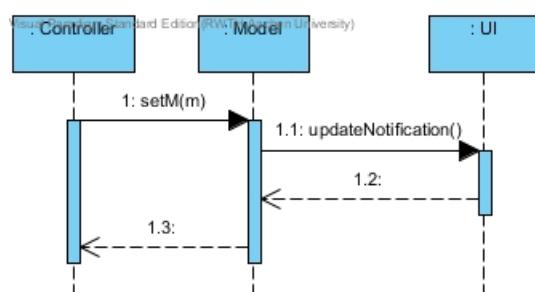


Abbildung 3.19: Sequenzdiagramm `setM`

setN

Das Sequenzdiagramm für *setN* ist in 3.20 dargestellt.

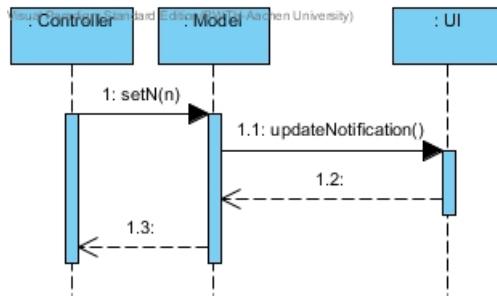


Abbildung 3.20: Sequenzdiagramm setN

setRightBoundary

Das Sequenzdiagramm für *setRightBoundary* ist in 3.21 dargestellt.

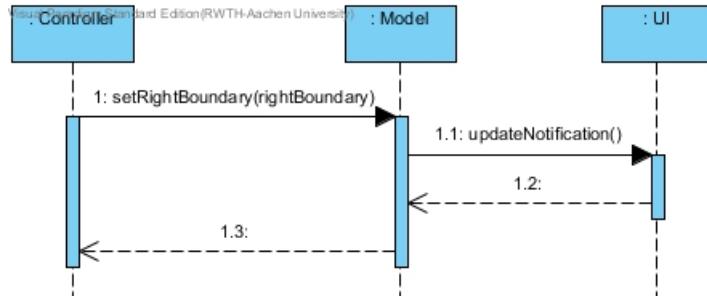


Abbildung 3.21: Sequenzdiagramm setRightBoundary

setT

Das Sequenzdiagramm für *setT* ist in 3.22 dargestellt.

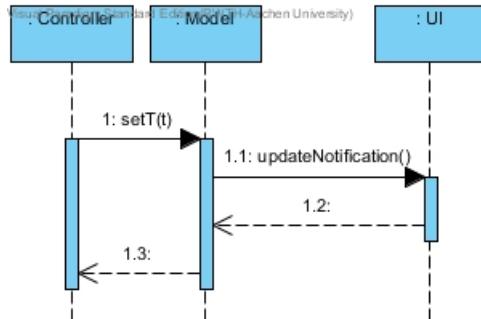


Abbildung 3.22: Sequenzdiagramm setT

setTopBoundary

Das Sequenzdiagramm für *setTopBoundary* ist in 3.23 dargestellt.

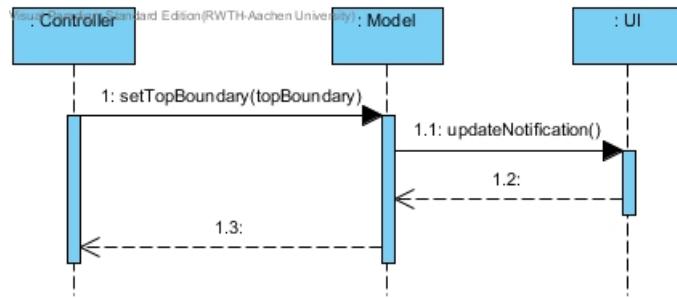


Abbildung 3.23: Sequenzdiagramm setTopBoundary

simulate

Das Sequenzdiagramm für *simulate* ist in 3.24 dargestellt.

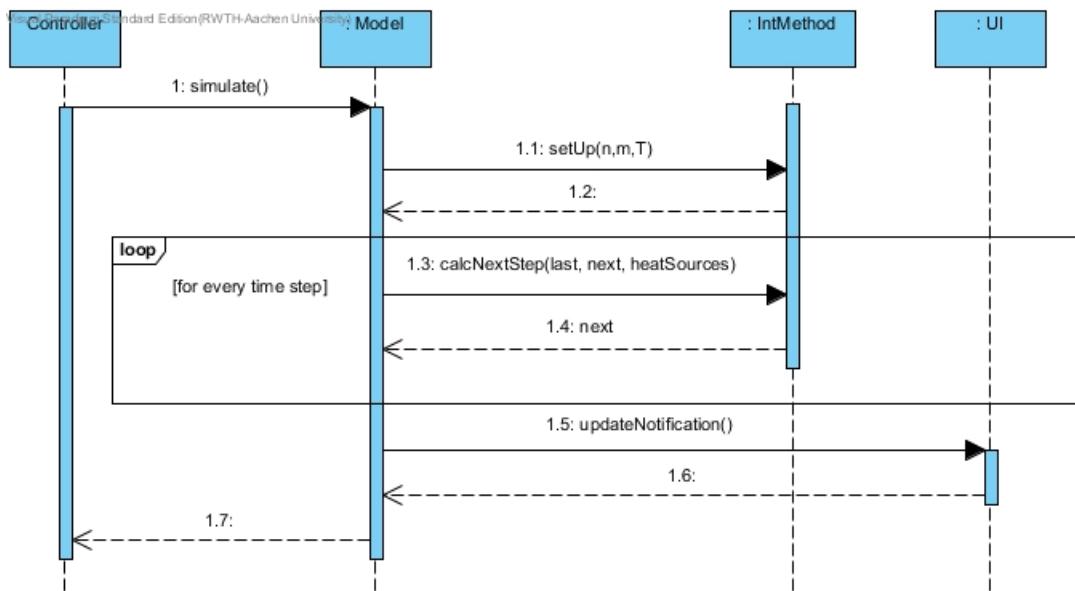


Abbildung 3.24: Sequenzdiagramm simulate

3.3.3 Paket presentation

Das Klassendiagramm in Abbildung 3.25 zeigt alle im Paket *presentation* enthaltene Klassen.

```

- activeTab : int
- model : Model&
- result : double*** &
- resultM : int&
- resultN : int&
- resultT : double&

+drawPartialHeatSource(partialAreaX : vector <double>&, partialAreaY : vector <double>&) : void
+drawPartialThermalConductivity(partialAreaX : vector <double>&, partialAreaY : vector <double>&) : void
+testFocusChange(old : QWidget *) : bool
+getInitialFrame() : int
+getNewHeatSourceValue(row : int) : double
+getNewThermalConductivityValue(row : int) : double
+heatSourcePixelToCoords(mouseX : double, mouseY : double, x : double &, y : double &) : void
+revertFocusChange(old : QWidget *, now : QWidget *) : void
+setActiveTab(tab : int) : void
+setController(controller : Controller &) : void
+setModel(model : Model &) : void
+thermalConductivityPixelToCoords(mouseX : double, mouseY : double, x : double &, y : double &) : void
-updateHeatSources() : void
-updateIBVs() : void
+updateNotification() : void
-updateSimulating() : void
-updateThermalConductivities() : void
-updateVisualization() : void
-visualizeHeatSourceArea(area : Area *) : void
+visualizeState(frame : int) : void
-visualizeThermalConductivityArea(area : Area *) : void

```

```

- model : mo
- partialArea
- partialArea
- startedNew
- startedNew
+ focusChan
+ heatSource
+ heatSource
+ newBottom
+ newInitialV
+ newLeftBo
+ newMSlot(
+ newNSlot(
+ newRightB
+ newTopBo
+ newTSlot(r
+ playVideoS
+ selectIntM
+ selectIterat
+ setModel(r
+ setUI(ui : U
+ simulateSk
+ testPartial
+ testPartial
+ thermalCon
+ thermalCon
+ undoHeatS
+ undoTherm
+ visualize St

```

Abbildung 3.25: Klassendiagramm presentation

3.3.3.1 UI

Es werden lediglich die Sequenzdiagramme der Update-Methoden dargestellt.

updateHeatSources

Das Sequenzdiagramm für *updateHeatSources* ist in 3.26 dargestellt.

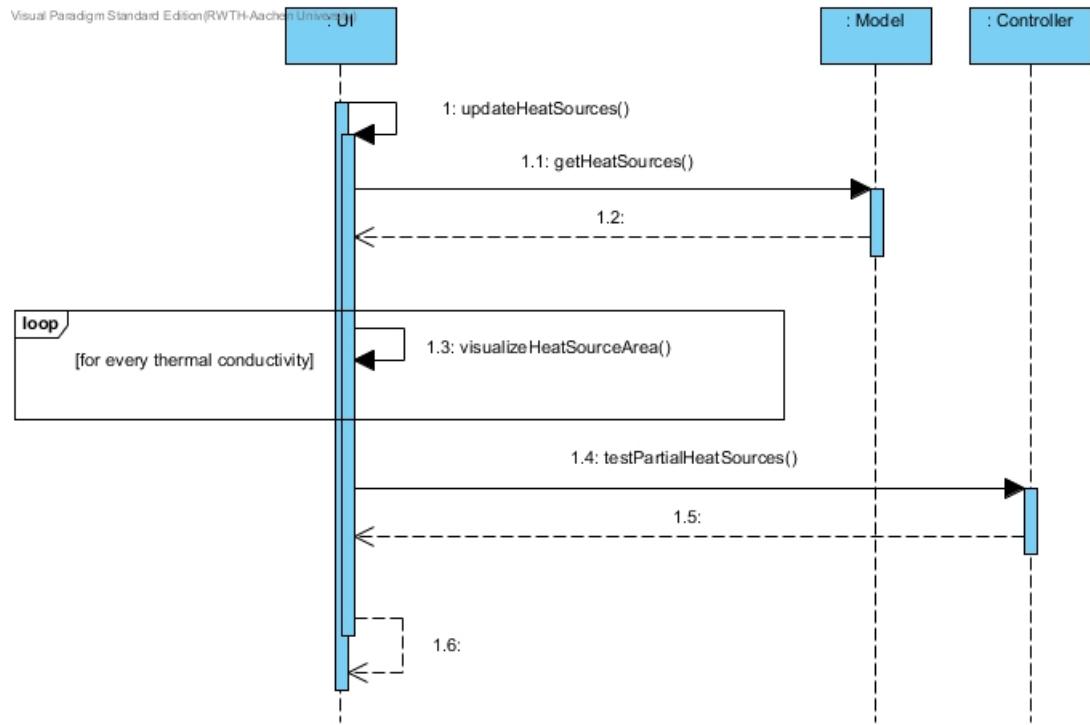


Abbildung 3.26: Sequenzdiagramm updateHeatSources

updateIBVs

Das Sequenzdiagramm für *updateIBVs* ist in 3.27 dargestellt.

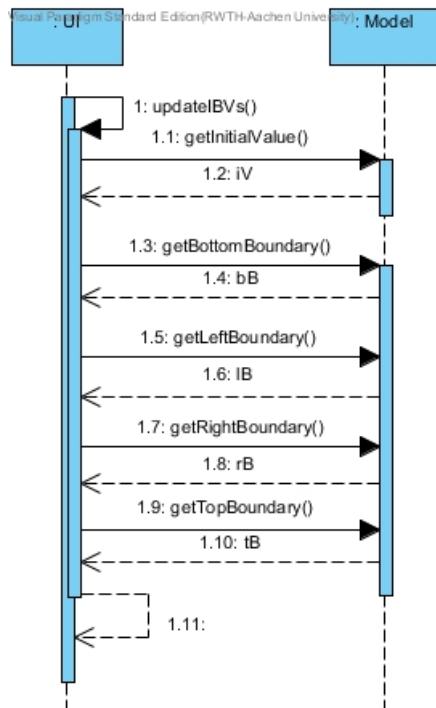


Abbildung 3.27: Sequenzdiagramm updateIBVs

updateNotification

Das Sequenzdiagramm für *updateNotification* ist in 3.28 dargestellt.

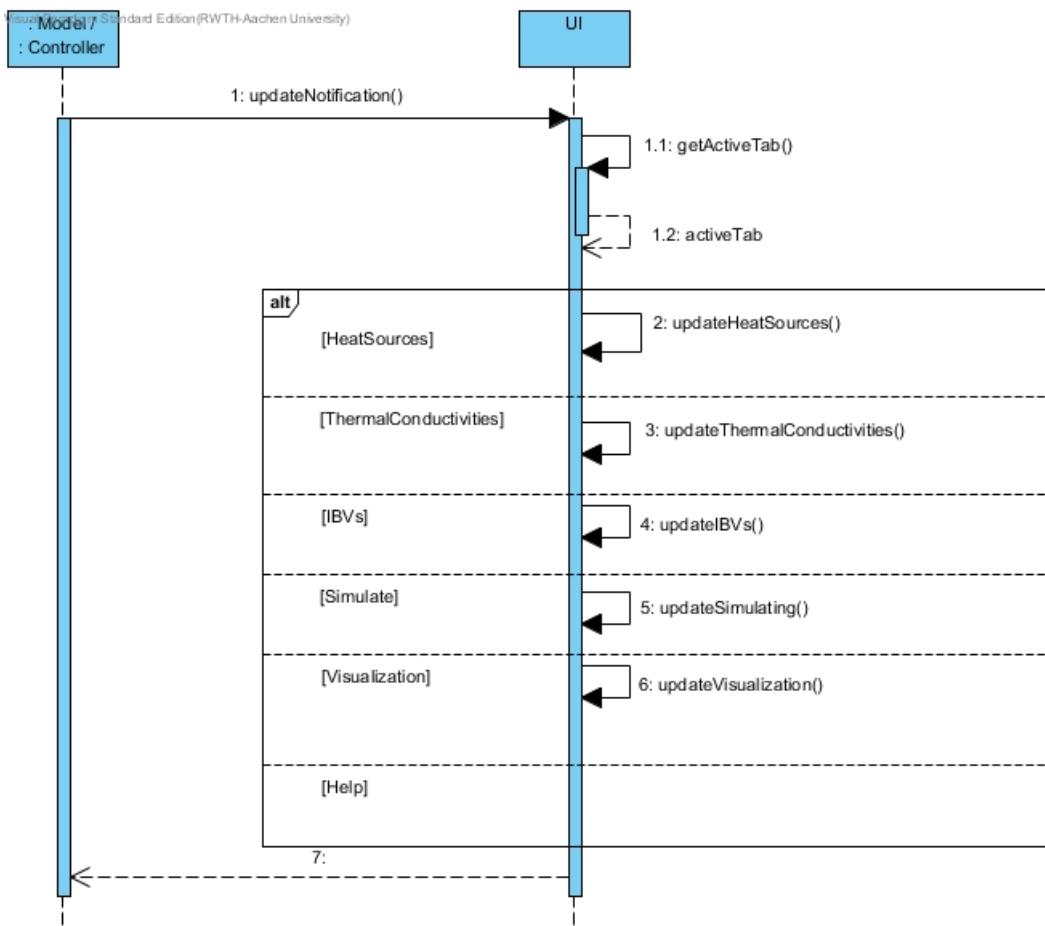


Abbildung 3.28: Sequenzdiagramm `updateNotification`

updateSimulating

Das Sequenzdiagramm für `updateSimulating` ist in 3.29 dargestellt.

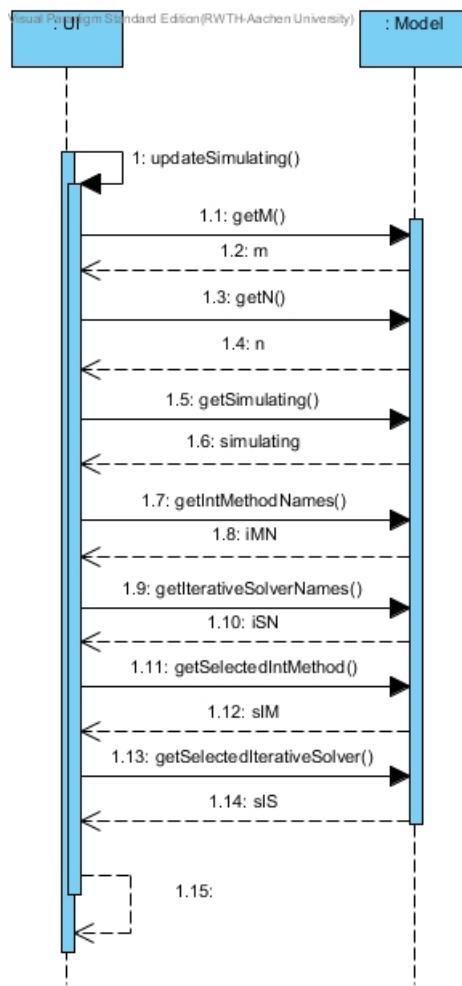


Abbildung 3.29: Sequenzdiagramm `updateSimulating`

updateThermalConductivities

Das Sequenzdiagramm für `updateThermalConductivities` ist in 3.30 dargestellt.

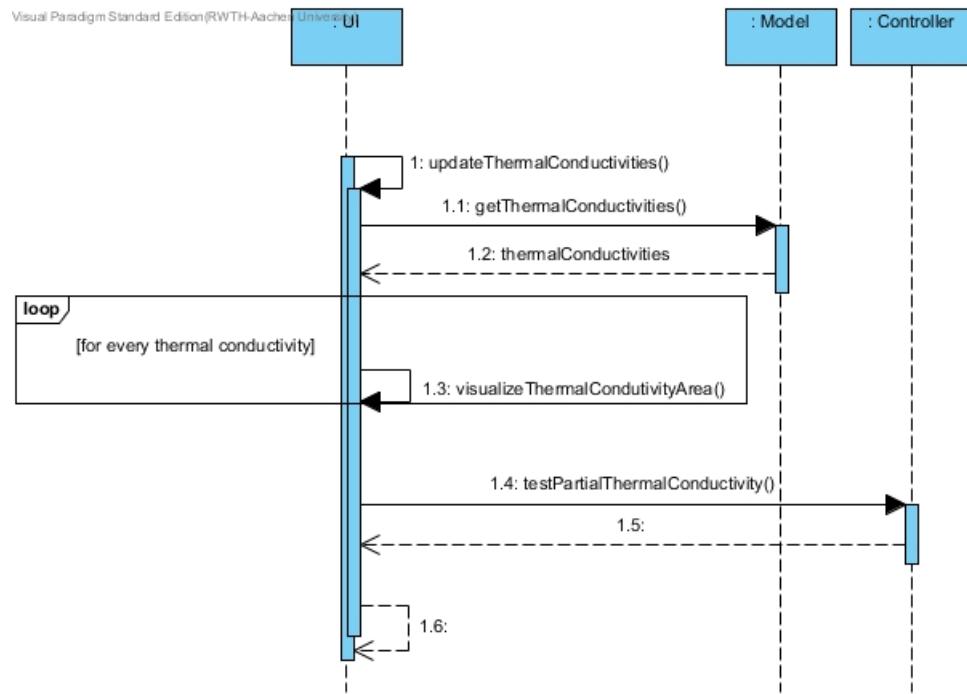


Abbildung 3.30: Sequenzdiagramm `updateThermalConductivities`

updateVisualization

Das Sequenzdiagramm für *updateVisualization* ist in 3.31 dargestellt.

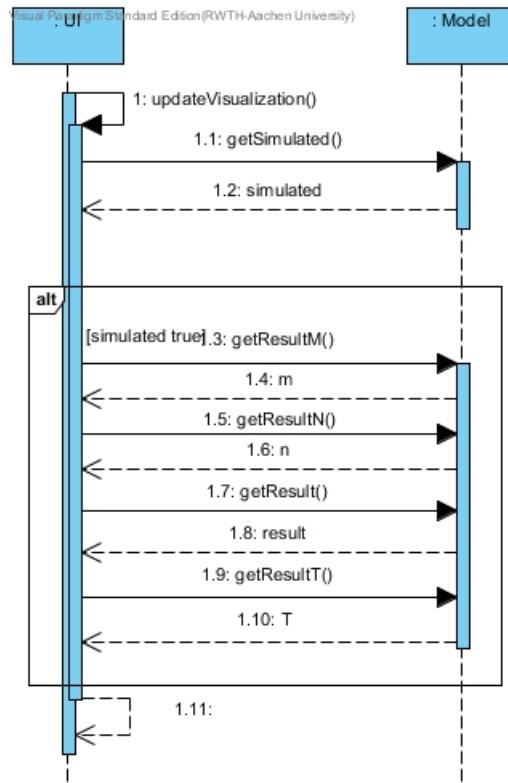


Abbildung 3.31: Sequenzdiagramm updateVisualization

3.3.3.2 Controller

focusChangedSlot

Das Sequenzdiagramm für *focusChangedSlot* ist in 3.32 dargestellt.

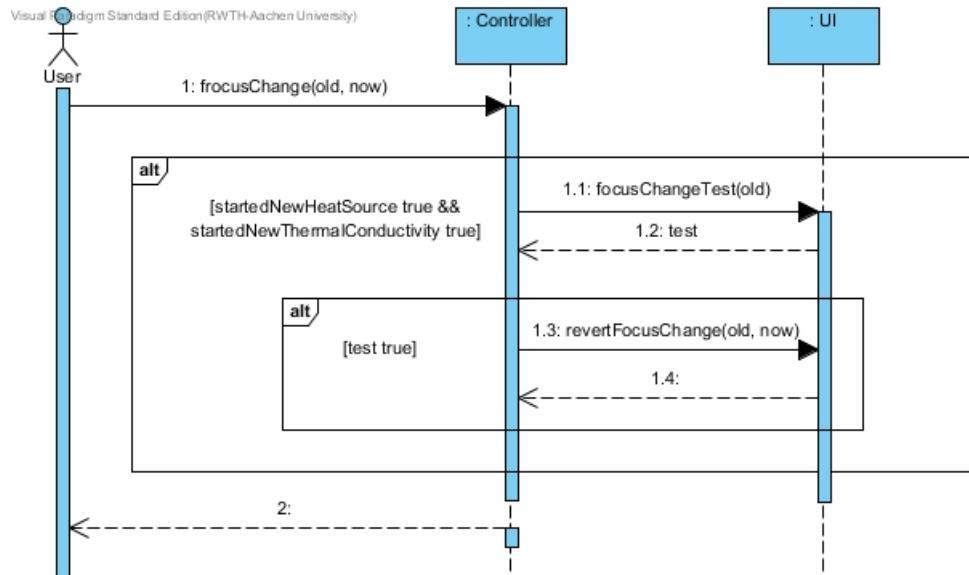


Abbildung 3.32: Sequenzdiagramm focusChangedSlot

heatSourceClickSlot

Das Sequenzdiagramm für *heatSourceClickSlot* ist in 3.33 dargestellt.

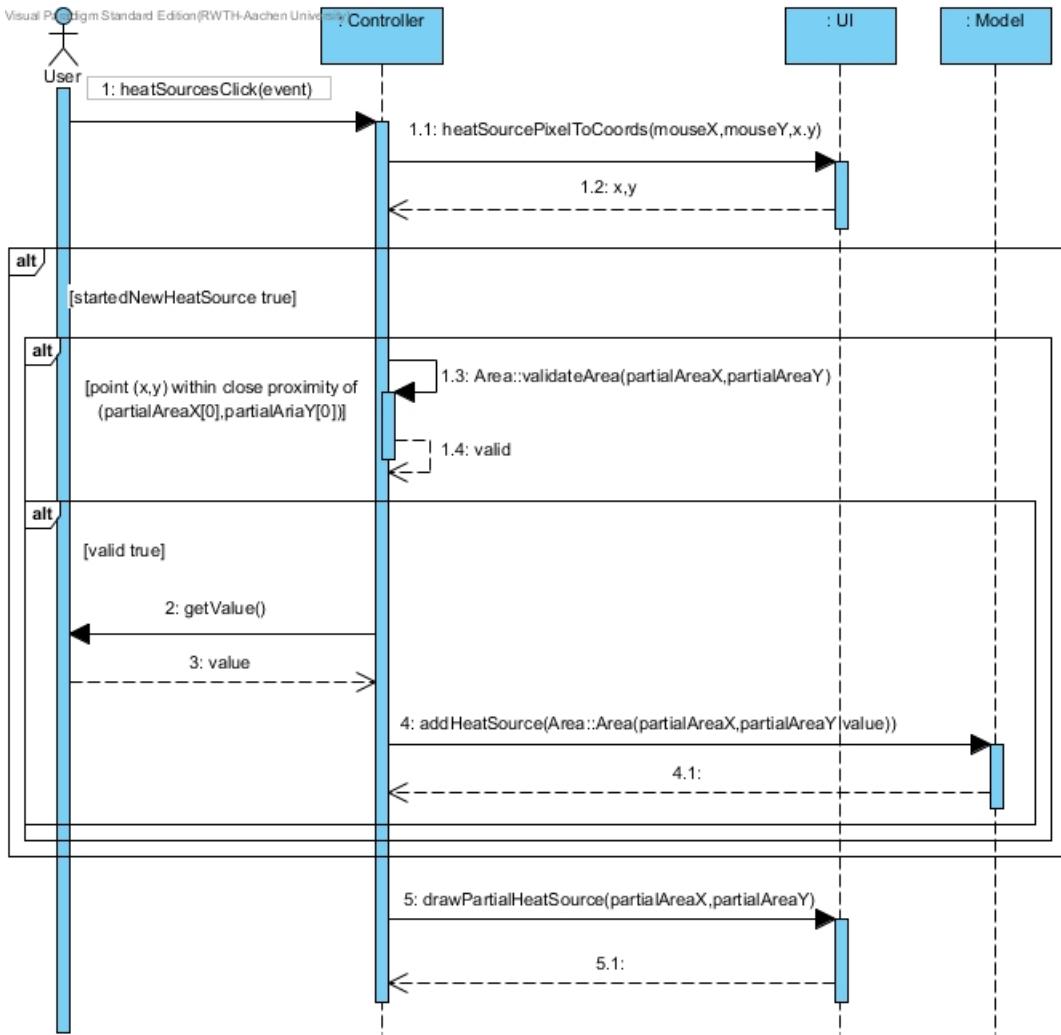


Abbildung 3.33: Sequenzdiagramm heatSourceClickSlot

newBottomBoundarySlot

Das Sequenzdiagramm für *newBottomBoundarySlot* ist in 3.34 dargestellt.

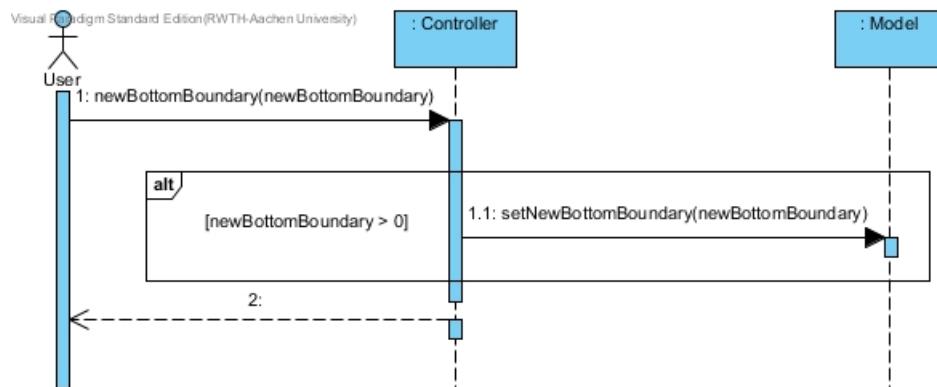


Abbildung 3.34: Sequenzdiagramm newBottomBoundarySlot

newInitialValueSlot

Das Sequenzdiagramm für *newInitialValueSlot* ist in 3.36 dargestellt.

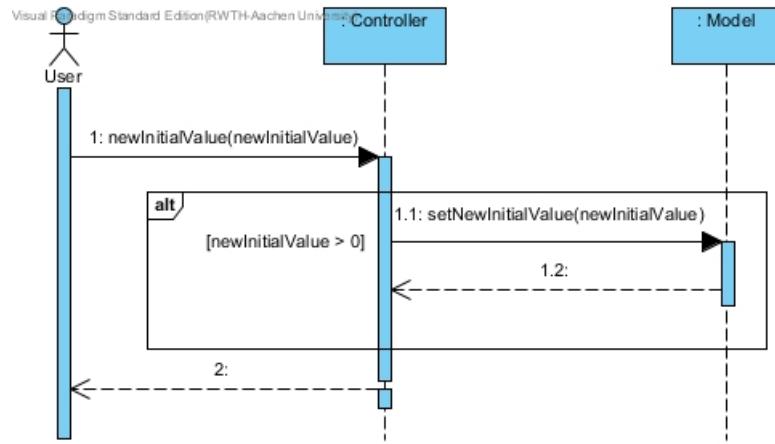


Abbildung 3.35: Sequenzdiagramm *newInitialValueSlot*

newInitialValueSlot

Das Sequenzdiagramm für *newInitialValueSlot* ist in 3.36 dargestellt.

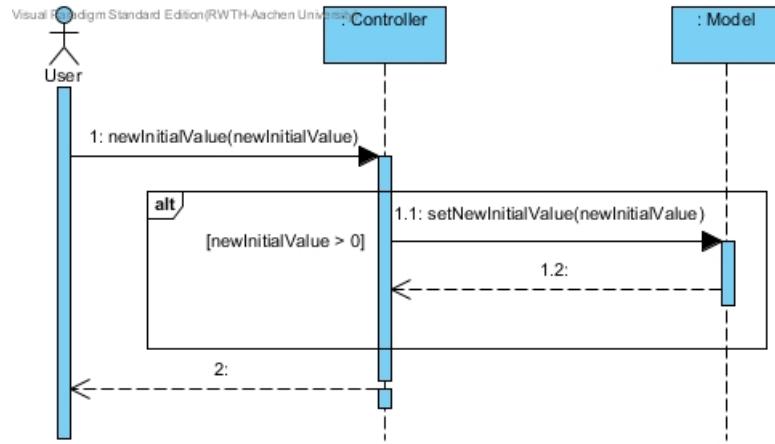


Abbildung 3.36: Sequenzdiagramm *newInitialValueSlot*

newLeftBoundarySlot

Das Sequenzdiagramm für *newLeftBoundarySlot* ist in 3.37 dargestellt.

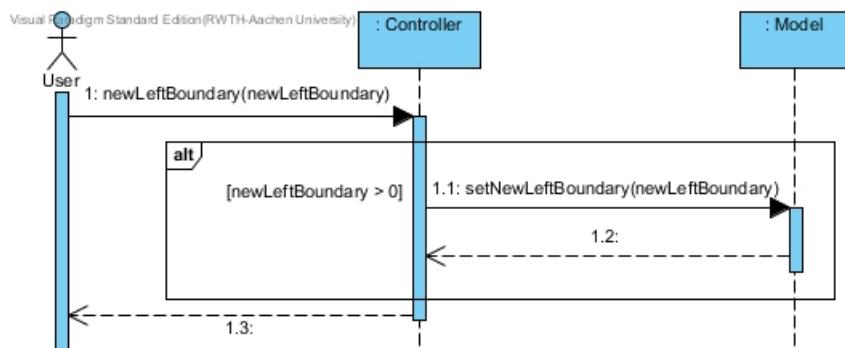


Abbildung 3.37: Sequenzdiagramm *newLeftBoundarySlot*

newMSlot

Das Sequenzdiagramm für *newMSlot* ist in 3.38 dargestellt.

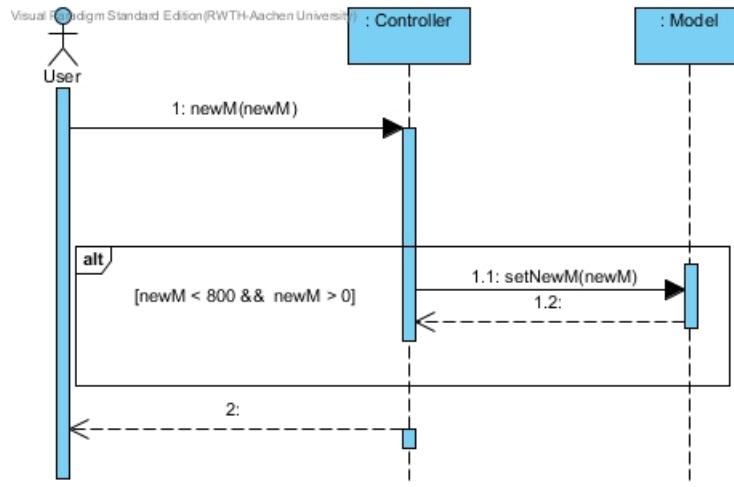


Abbildung 3.38: Sequenzdiagramm newMSlot

newNSlot

Das Sequenzdiagramm für *newNSlot* ist in 3.39 dargestellt.

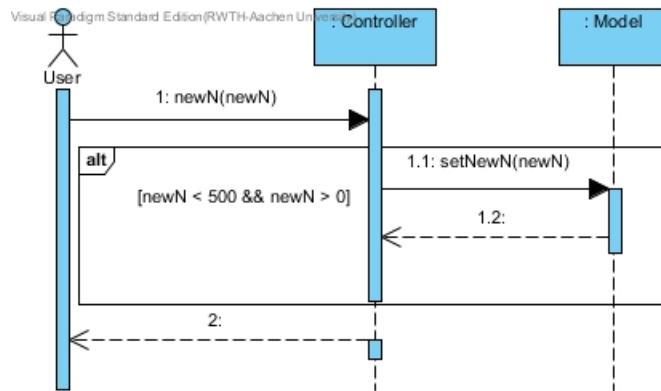


Abbildung 3.39: Sequenzdiagramm newNSlot

newRightBoundarySlot

Das Sequenzdiagramm für *newRightBoundarySlot* ist in 3.40 dargestellt.

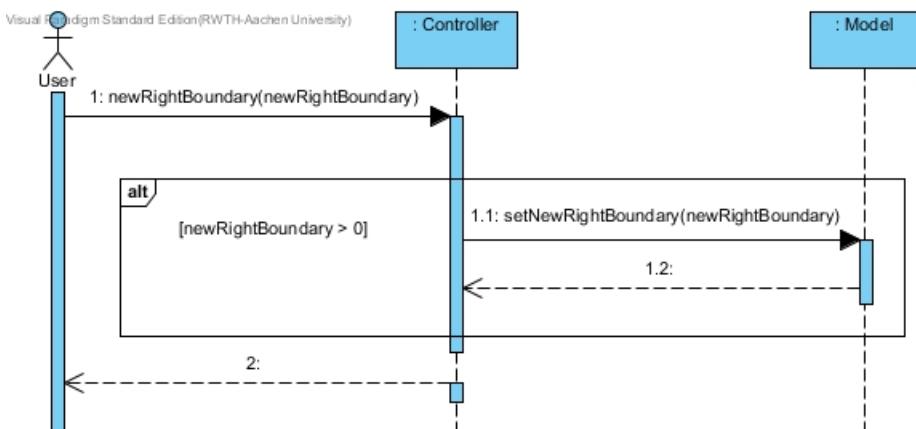


Abbildung 3.40: Sequenzdiagramm newRightBoundarySlot

newTopBoundarySlot

Das Sequenzdiagramm für *newTopBoundarySlot* ist in 3.41 dargestellt.

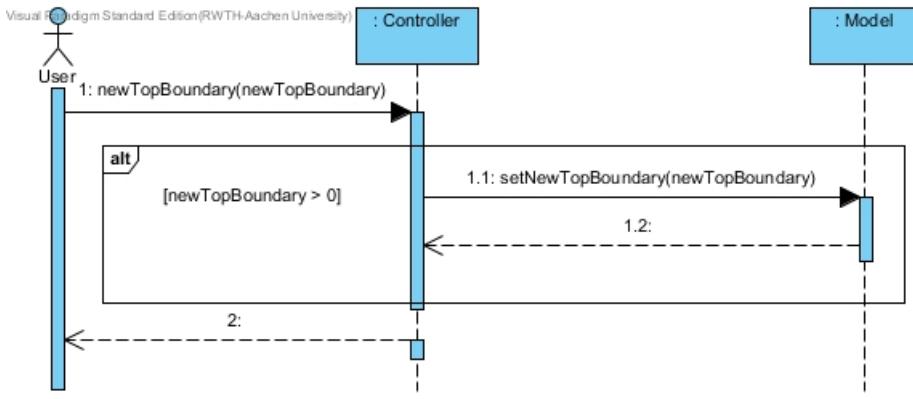


Abbildung 3.41: Sequenzdiagramm *newTopBoundarySlot*

newTSlot

Das Sequenzdiagramm für *newTSlot* ist in 3.43 dargestellt.

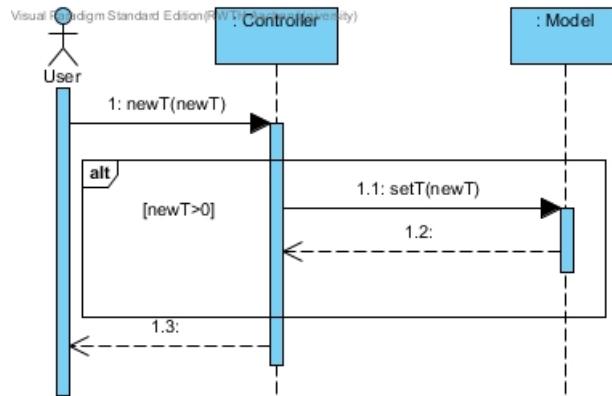


Abbildung 3.42: Sequenzdiagramm *newTSlot*

newTSlot

Das Sequenzdiagramm für *newTSlot* ist in 3.43 dargestellt.

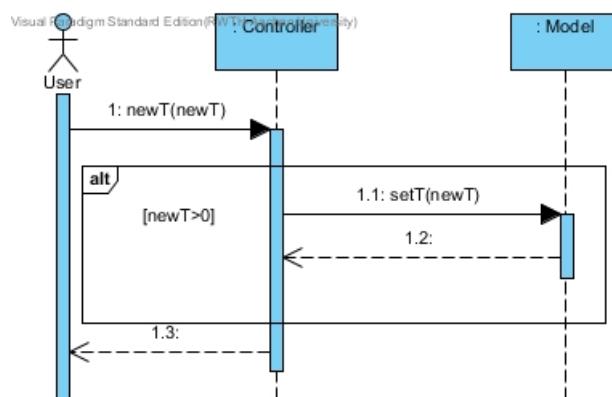


Abbildung 3.43: Sequenzdiagramm *newTSlot*

playVideoSlot

Das Sequenzdiagramm für *playVideoSlot* ist in 3.44 dargestellt.

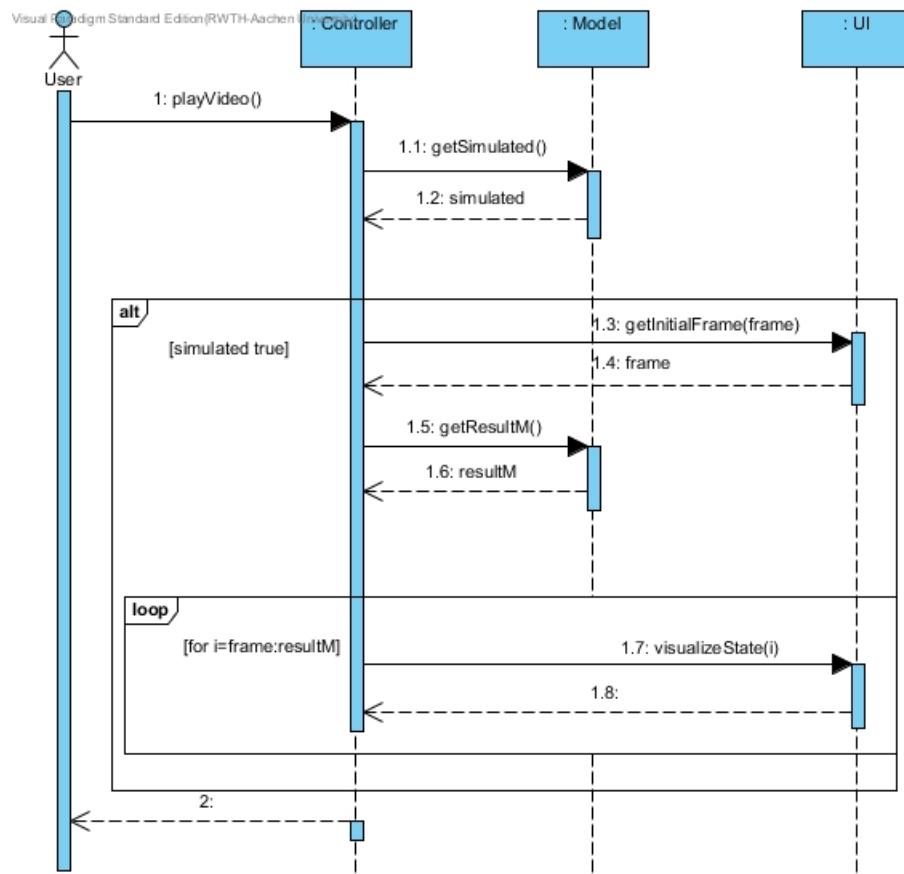


Abbildung 3.44: Sequenzdiagramm `playVideoSlot`

selectIntMethodSlot

Das Sequenzdiagramm für `selectIntMethodSlot` ist in 3.45 dargestellt.

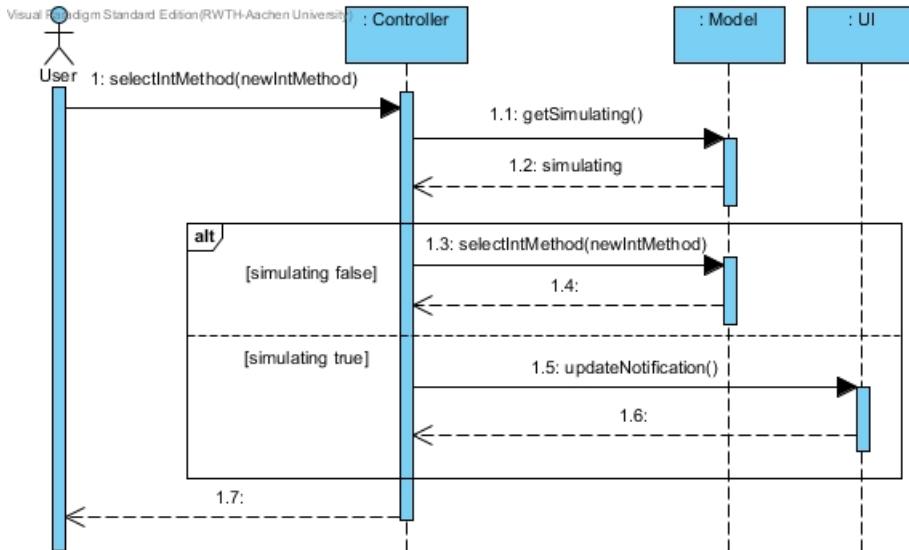


Abbildung 3.45: Sequenzdiagramm `selectIntMethodSlot`

selectIterativeSolverSlot

Das Sequenzdiagramm für `selectIterativeSolverSlot` ist in 3.46 dargestellt.

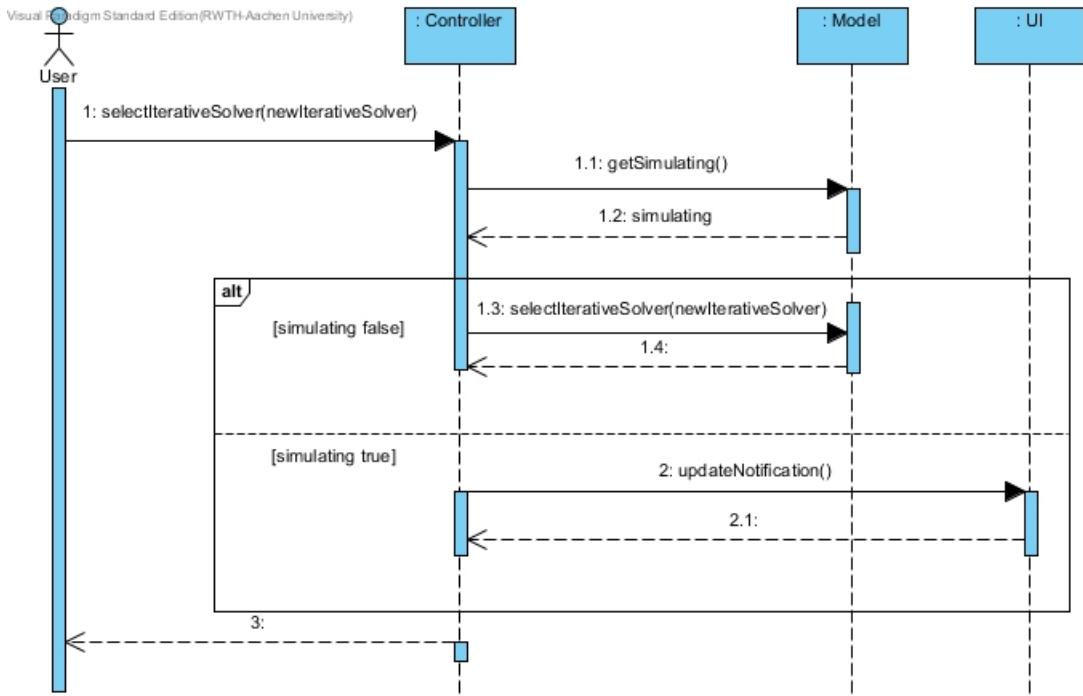


Abbildung 3.46: Sequenzdiagramm selectIterativeSolverSlot

simulateSlot

Das Sequenzdiagramm für *simulateSlot* ist in 3.47 dargestellt.

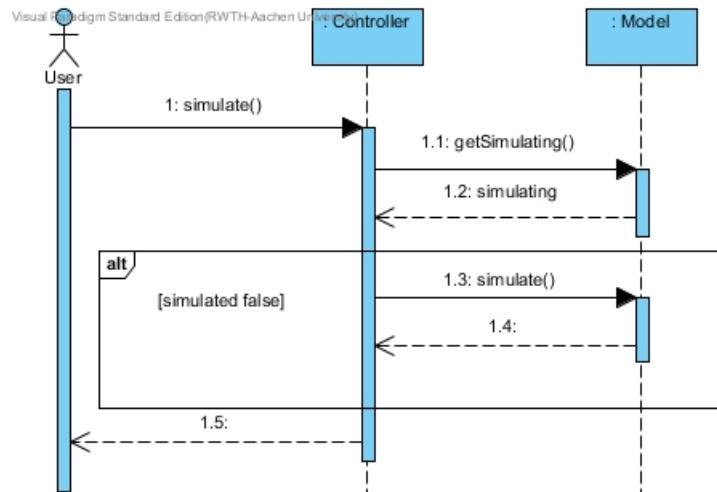


Abbildung 3.47: Sequenzdiagramm simulateSlot

thermalConductivitiesClickSlot

Das Sequenzdiagramm für *thermalConductivitiesClickSlot* ist in 3.48 dargestellt.

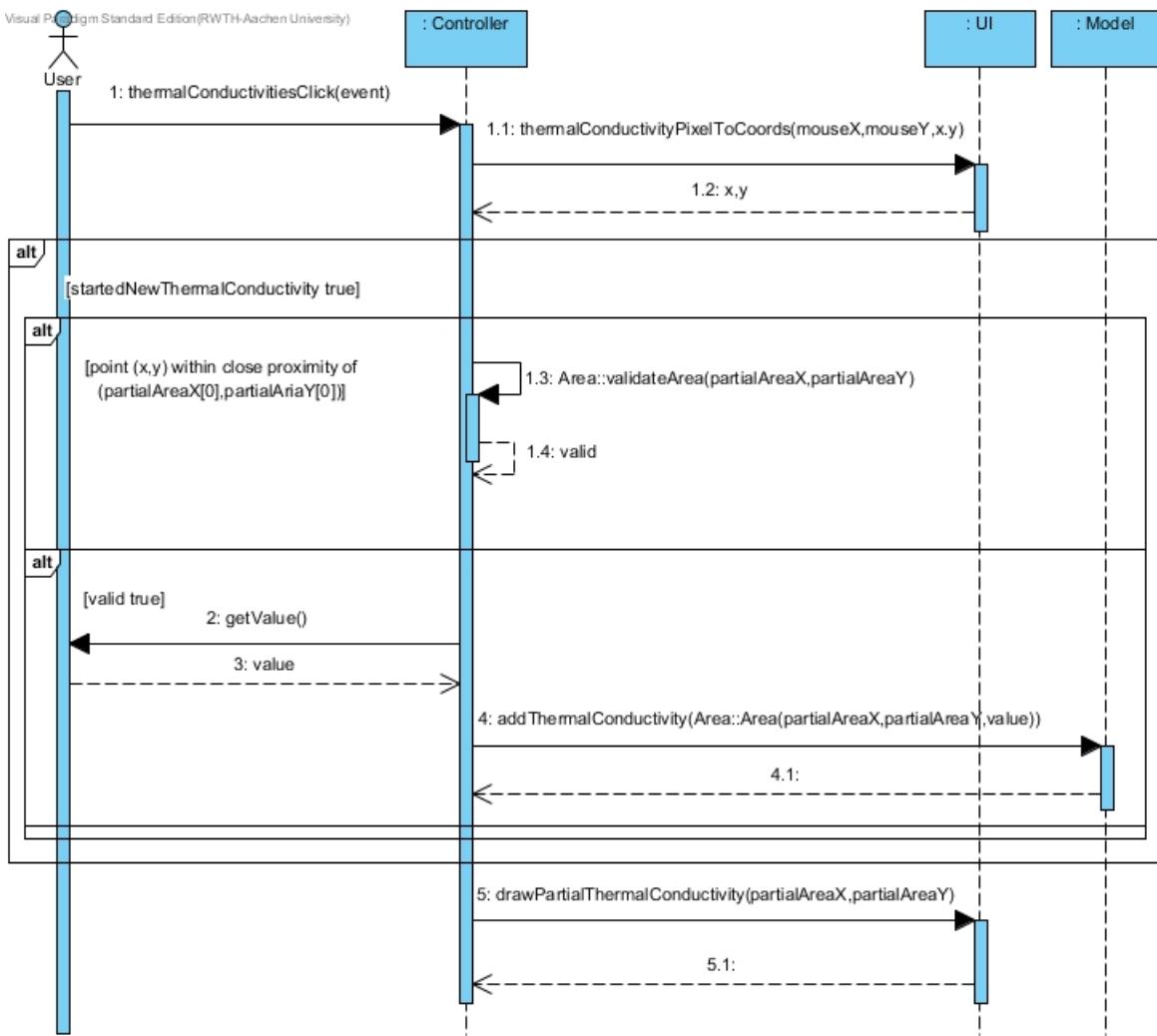


Abbildung 3.48: Sequenzdiagramm thermalConductivitiesClickSlot

undoHeatSourceSlot

Das Sequenzdiagramm für *undoHeatSourceSlot* ist in 3.49 dargestellt.

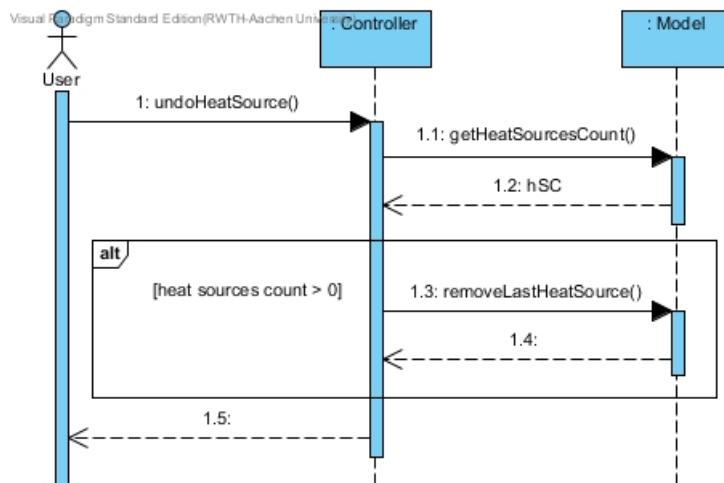


Abbildung 3.49: Sequenzdiagramm undoHeatSourceSlot

undoThermalConductivitySlot

Das Sequenzdiagramm für *undoThermalConductivitySlot* ist in 3.50 dargestellt.

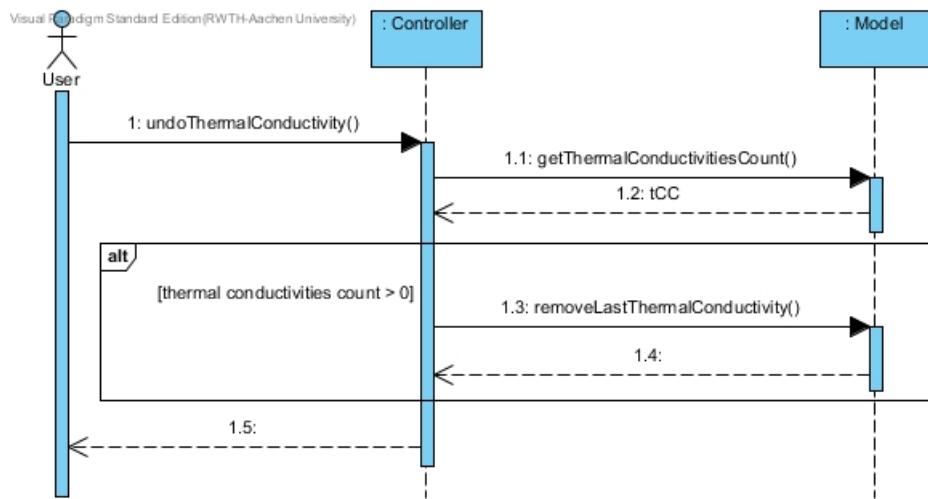


Abbildung 3.50: Sequenzdiagramm *undoThermalConductivitySlot*

visualizeStateSlot

Das Sequenzdiagramm für *visualizeStateSlot* ist in 3.51 dargestellt.

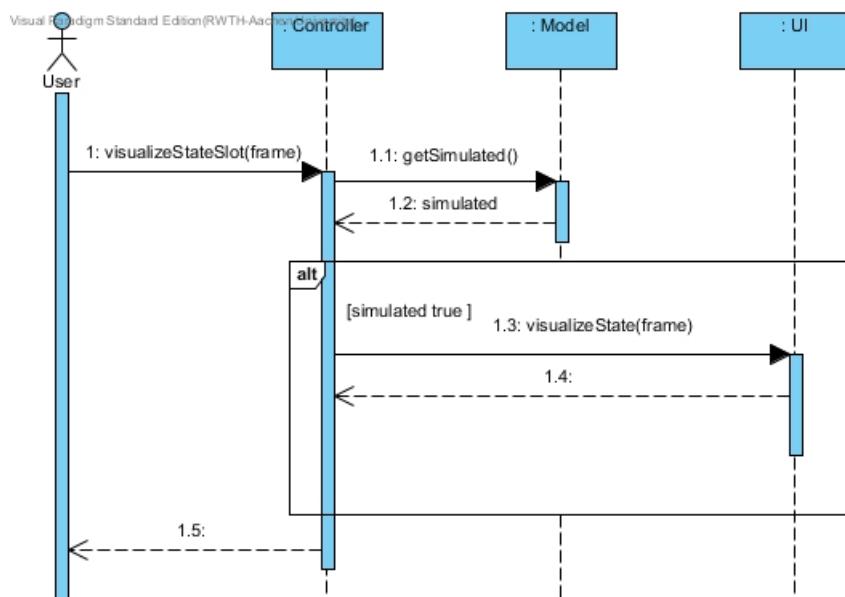


Abbildung 3.51: Sequenzdiagramm *visualizeStateSlot*

Kapitel 4

Entwicklerdokumentation

4.1 Codestruktur

4.2 Detaillierte Dokumentation des Codes

4.3 Software Tests

Zur Verifikation der Software haben wir verschiedene Testmethoden verwendet. Im Folgenden werden diese Tests nach Code-Zugehörigkeit sortiert aufgelistet:

Die Simulationsergebnisse der Software wurden unter Auswahl der Kombination "Impliziter Euler - Jacobi" gegen einen im Zuge der Veranstaltung "Mathematische Grundlagen IV" verifizierten Matlab-Simulationscode getestet, der zugehörige Code wird im Anhang präsentiert. Die weiteren numerischen Methoden wurden anschließend gegen dieses Referenzergebnis getestet.

4.3.1 algorithms

4.3.1.1 CRS

Die folgenden Funktionalitäten wurden vom Programmierer per Ausgabe (Vergleich Zahlenwerte) verifiziert:

1. A1
2. diag
3. diffCRS
4. full
5. eye
6. multCRSCRS
7. multCRSQVector
8. multRowQVector
9. multRowQVectorAbs
10. scalarCRS
11. scalarQVector
12. sumCRS
13. sumQVector

4.3.2 Solver

Die folgenden Löser per Ausgabe (Vergleich Zahlenwerte) anhand von Beispielsystemen verifiziert:

1. Jacobi
2. Gauss-Seidel
3. LU (direkter Löser)

Abbildungsverzeichnis

2.1	Anwendungsfalldiagramm	3
2.2	Aktivitätsdiagramm Use Case Anfangsbedingungen eingeben	10
2.3	Aktivitätsdiagramm Use Case Diskretisierungsgrößen eingeben	10
2.4	Aktivitätsdiagramm Use Case Messungen eingeben	10
2.5	Aktivitätsdiagramm Use Case Messungen importieren	11
2.6	Aktivitätsdiagramm Use Case Randbedingungen eingeben	11
2.7	Aktivitätsdiagramm Use Case Parameter optimieren	11
2.8	Aktivitätsdiagramm Use Case Simulationsexperiment laden/speichern	12
2.9	Aktivitätsdiagramm Use Case Simulieren	12
2.10	Aktivitätsdiagramm Use Case Temperaturleitkoeffizienten eingeben	13
2.11	Aktivitätsdiagramm Use Case Video abspielen	13
2.12	Aktivitätsdiagramm Use Case Wärmequellen eingeben	14
2.13	Aktivitätsdiagramm Use Case Zustand anzeigen	14
2.14	Begriffsnetz	16
3.1	Paketstruktur	17
3.2	Klassendiagramm algorithms	18
3.3	Sequenzdiagramm calcNextStep	18
3.4	Sequenzdiagramm selectSolver	19
3.5	Sequenzdiagramm setUp	19
3.6	Sequenzdiagramm setUpSpecific	20
3.7	Sequenzdiagramm setEps	20
3.8	Sequenzdiagramm setMaxIt	21
3.9	Klassendiagramm model	22
3.10	Sequenzdiagramm addHeatSource	23
3.11	Sequenzdiagramm addThermalDiffusivity	23
3.12	Sequenzdiagramm removeLastHeatSource	23
3.13	Sequenzdiagramm removeLastThermalDiffusivity	24
3.14	Sequenzdiagramm selectIntMethod	24
3.15	Sequenzdiagramm selectSolver	24
3.16	Sequenzdiagramm setBottomBoundary	25
3.17	Sequenzdiagramm setInitialValue	25
3.18	Sequenzdiagramm setLeftBoundary	25
3.19	Sequenzdiagramm setM	25
3.20	Sequenzdiagramm setN	26
3.21	Sequenzdiagramm setRightBoundary	26
3.22	Sequenzdiagramm setT	26
3.23	Sequenzdiagramm setTopBoundary	27
3.24	Sequenzdiagramm simulate	27
3.25	Klassendiagramm presentation	28
3.26	Sequenzdiagramm updateHeatSources	29
3.27	Sequenzdiagramm updateIBVs	29
3.28	Sequenzdiagramm updateNotification	30

3.29	Sequenzdiagramm updateSimulating	31
3.30	Sequenzdiagramm updateThermalConductivities	31
3.31	Sequenzdiagramm updateVisualization	32
3.32	Sequenzdiagramm focusChangedSlot	32
3.33	Sequenzdiagramm heatSourceClickSlot	33
3.34	Sequenzdiagramm newBottomBoundarySlot	33
3.35	Sequenzdiagramm newInitialValueSlot	34
3.36	Sequenzdiagramm newInitialValueSlot	34
3.37	Sequenzdiagramm newLeftBoundarySlot	34
3.38	Sequenzdiagramm newMSlot	35
3.39	Sequenzdiagramm newNSlot	35
3.40	Sequenzdiagramm newRightBoundarySlot	35
3.41	Sequenzdiagramm newTopBoundarySlot	36
3.42	Sequenzdiagramm newTSlot	36
3.43	Sequenzdiagramm newTSlot	36
3.44	Sequenzdiagramm playVideoSlot	37
3.45	Sequenzdiagramm selectIntMethodSlot	37
3.46	Sequenzdiagramm selectIterativeSolverSlot	38
3.47	Sequenzdiagramm simulateSlot	38
3.48	Sequenzdiagramm thermalConductivitiesClickSlot	39
3.49	Sequenzdiagramm undoHeatSourceSlot	39
3.50	Sequenzdiagramm undoThermalConductivitySlot	40
3.51	Sequenzdiagramm visualizeStateSlot	40

Tabellenverzeichnis

2.1 Beschreibung Use Case Anfangsbedingungen eingeben	4
2.2 Beschreibung Use Case Diskretisierungsgrößen eingeben	4
2.3 Beschreibung Use Case Messungen eingeben	5
2.4 Beschreibung Use Case Messungen importieren	5
2.5 Beschreibung Use Case Randbedingungen eingeben	6
2.6 Beschreibung Use Case Parameter optimieren	6
2.7 Beschreibung Use Case Simulationsexperiment laden/speichern	7
2.8 Beschreibung Use Case Simulieren	7
2.9 Beschreibung Use Case Temperaturleitkoeffizienten eingeben	8
2.10 Beschreibung Use Case Video abspielen	8
2.11 Beschreibung Use Case Wärmequellen eingeben	9
2.12 Beschreibung Use Case Zustand anzeigen	9