

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

<b>Inhaltsverzeichnis</b>	<b>2</b>
<b>1 Vorwort</b>	<b>3</b>
1.1 Aufgabenstellung und Struktur des Dokument . . . . .	3
1.2 Projektmanagement . . . . .	3
1.3 Lob und Kritik . . . . .	3
<b>2 Analyse</b>	<b>4</b>
2.1 Anforderungsanalyse . . . . .	4
2.1.1 Benutzeranforderungen . . . . .	4
2.1.2 Anwendungsfallanalyse . . . . .	4
2.1.2.1 Anwendungsfalldiagramm . . . . .	4
2.1.2.2 Beschreibungen der Anwendungsfälle . . . . .	6
2.1.2.3 Aktivitätsdiagramme . . . . .	11
2.1.2.4 Systemanforderungen . . . . .	16
2.2 Begriffsanalyse . . . . .	17
2.2.1 Klassenkandidaten . . . . .	17
2.2.2 Begriffsnetz . . . . .	18
<b>3 Entwurf</b>	<b>19</b>
3.1 Pakete . . . . .	19
3.2 Klassen . . . . .	20
3.2.1 Paket algorithms . . . . .	20
3.2.1.1 IntMethod . . . . .	21
3.2.2 Paket model . . . . .	22
3.2.2.1 Model . . . . .	23
3.2.2.2 SimulationSetup . . . . .	36
3.2.2.3 SimulationWorker . . . . .	37
3.2.3 Paket presentation . . . . .	40
3.2.3.1 AreaWidget . . . . .	40
3.2.3.2 Controller . . . . .	42
3.2.3.3 UI . . . . .	56
<b>4 Benutzerdokumentation</b>	<b>61</b>
4.1 Installation . . . . .	61
4.1.1 Linux . . . . .	61
4.1.2 Windows . . . . .	61
4.1.3 Doxygen . . . . .	61
4.1.3.1 Linux . . . . .	61
4.1.3.2 Windows . . . . .	61
4.2 Beispielsitzung . . . . .	62
4.3 Fehlersituationen . . . . .	70

<b>5 Entwicklerdokumentation</b>	<b>71</b>
5.1 Codestruktur . . . . .	71
5.2 Detaillierte Dokumentation des Codes . . . . .	71
5.3 Software Tests . . . . .	71
5.3.1 presentation . . . . .	71
5.3.2 model . . . . .	72
5.3.2.1 Area . . . . .	72
5.3.3 algorithms . . . . .	72
5.3.3.1 CRS . . . . .	72
5.3.3.2 Solver . . . . .	72
<b>A Anhang</b>	<b>73</b>
A.1 Matlab-Code . . . . .	73
<b>Abbildungsverzeichnis</b>	<b>78</b>
<b>Tabellenverzeichnis</b>	<b>79</b>

# Kapitel 1

## Vorwort

Dieses Dokument erfasst den Verlauf des *Simulationssoftwareentwicklungspraktikums* des Studiengangs Computational Engineering Science an der RWTH Aachen im Jahr 2015.

Gegenüber der Präsentation wurden noch folgende Dinge ergänzt:

Eine modifizierte Version des Use-Case-Diagramms wurde im Abschnitt 2.1.2.1 *Anwendungsfalldiagramm* auf Seite 5 ergänzt. Des Weiteren wurden zusätzlich eine Visualisierung der Optimierungsergebnisse hinsichtlich der örtlichen Verteilung der Temperaturleitkoeffizienten sowie die Möglichkeit zur Übernahme der Ergebnisse in das aktuelle Simulationsexperiment implementiert.

### 1.1 Aufgabenstellung und Struktur des Dokument

Aufgabe ist es eine Software zur Simulation eines 2D-Wärmeleitungsproblems zu erstellen. Die Struktur des Dokumentes orientiert sich am Entwicklungsprozess der Software, beginnt mit der Analyse des Problems, fährt fort mit dem Entwurf der Software, woran sich die Benutzer- und zuletzt die Entwicklerdokumentation anschließt.

### 1.2 Projektmanagement

Die Problemanalyse sowie der Entwurf der Software erfolgten in Gruppenarbeit. Während der Implementierungsphase wurden die Aufgaben, grob an der Paketstruktur orientiert, an die einzelnen Mitglieder verteilt. Alle sich anschließenden Schritte wurden gemeinsam bearbeitet.

### 1.3 Lob und Kritik

Ein ausdrückliches Dankeschön möchten wir Markus Towara aussprechen, der uns stets umgehend bei Fragen und Problemen, ob persönlich oder per E-Mail, geholfen hat. Als Verbesserungsvorschlag: Ein beispielhafter Datensatz experimenteller Daten könnte gestellt werden, um das Parameter-Fitting besser nachvollziehen/verifizieren zu können.

# 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

Einen ersten Entwurf des Anwendungsfalldiagramm zeigt die Abbildung 2.1.

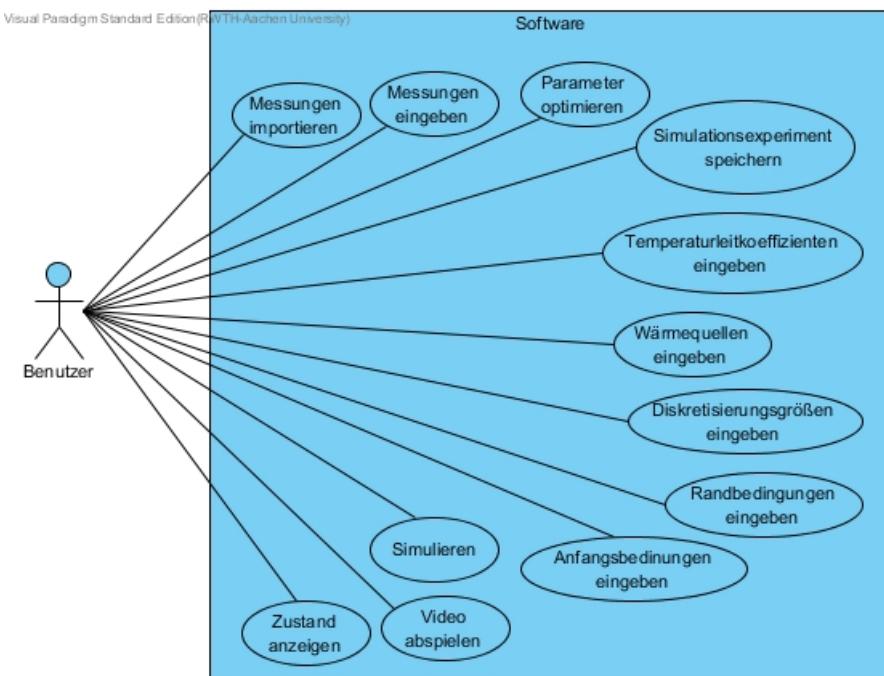


Abbildung 2.1: Anwendungsfalldiagramm

Eine modifizierte Version des Anwendungsfalldiagramms zeigt die Abbildung 2.2, das zur Grundlage der nächsten Iteration des Softwareentwicklungsprozesses dienen könnte.

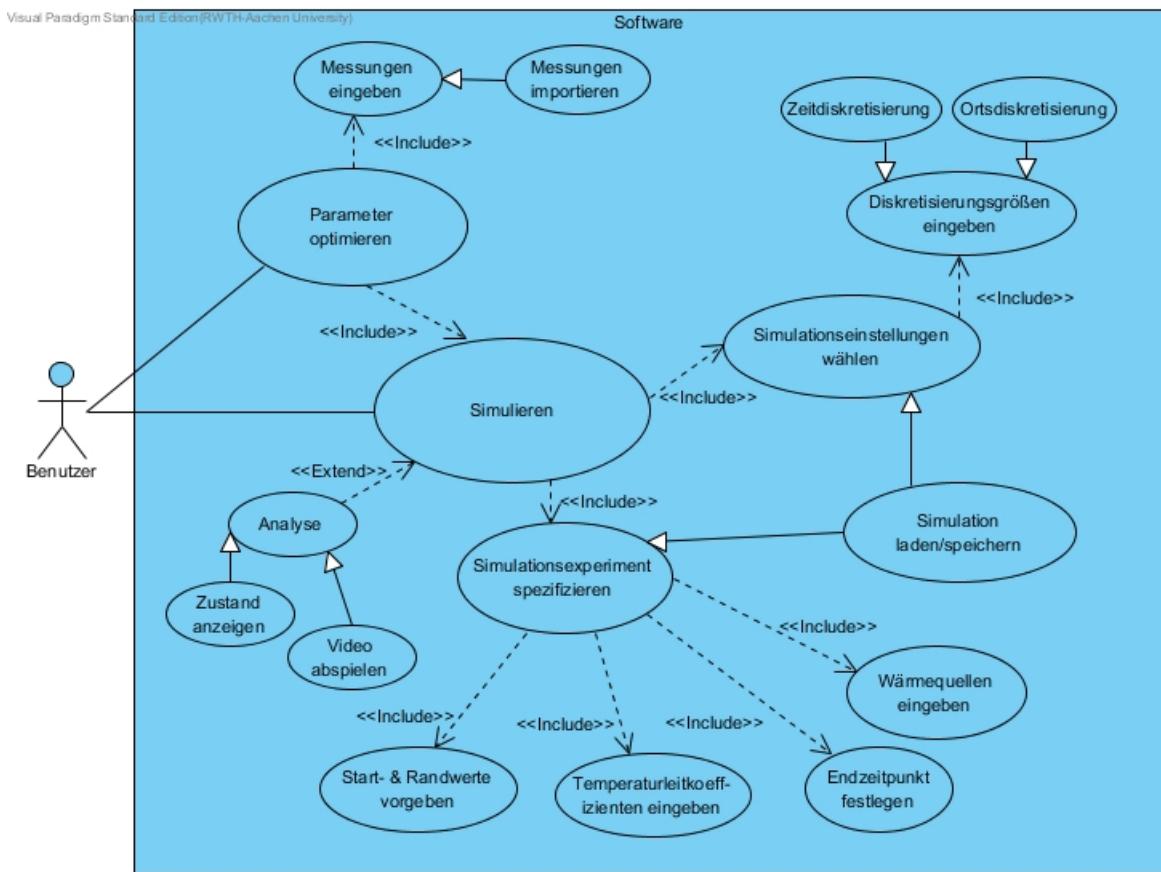


Abbildung 2.2: modifiziertes Anwendungsfalldiagramm

### 2.1.2.2 Beschreibungen der Anwendungsfälle

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

<b>Name</b>	Anfangsbedingungen eingeben	
<b>Ziel</b>	Der Benutzer möchte Anfangsbedingungen vorgeben.	
<b>Einordnung</b>	Basisfunktion	
<b>Vorbedingung</b>	Die Software wird korrekt ausgeführt.	
<b>Nachbedingung</b>	Die Anfangsbedingungen wurden vorgegeben und gespeichert.	
<b>Nachbedingung im Fehlerfall</b>	Die Anfangsbedingungen wurden nicht geändert und entsprechende Fehlermeldungen wurden ausgegeben.	
<b>Haupt-Neben-Akteur</b>	Benutzer	
<b>Auslöser</b>	Der Benutzer möchte Anfangsbedingungen vorgeben.	
<b>Standardfluss</b>	<b>Schritt</b>	<b>Aktion</b>
	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.
<b>Nebenfluss</b>	<b>Schritt</b>	<b>Aktion</b>
<b>Anfangsbedingungen nicht akzeptiert</b>	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

<b>Name</b>	Diskretisierungsgrößen eingeben	
<b>Ziel</b>	Der Benutzer möchte Diskretisierungsgrößen eingeben.	
<b>Einordnung</b>	Basisfunktion	
<b>Vorbedingung</b>	Die Software wird korrekt ausgeführt.	
<b>Nachbedingung</b>	Die Diskretisierungsgrößen wurden vorgegeben und gespeichert.	
<b>Nachbedingung im Fehlerfall</b>	Die Diskretisierungsgrößen wurden nicht geändert und entsprechende Fehlermeldungen wurden ausgegeben.	
<b>Haupt-Neben-Akteur</b>	Benutzer	
<b>Auslöser</b>	Der Benutzer möchte Diskretisierungsgrößen eingeben.	
<b>Standardfluss</b>	<b>Schritt</b>	<b>Aktion</b>
	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.
<b>Nebenfluss</b>	<b>Schritt</b>	<b>Aktion</b>
<b>Eingegebene Größen nicht akzeptiert</b>	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

<b>Name</b>	Messungen eingeben	
<b>Ziel</b>	Der Benutzer möchte experimentelle Daten eingeben.	
<b>Einordnung</b>	Basisfunktion	
<b>Vorbedingung</b>	Die Software wird korrekt ausgeführt.	
<b>Nachbedingung</b>	Die eingegebenen Daten wurden gespeichert.	
<b>Nachbedingung im Fehlerfall</b>	Die Daten wurden nicht geändert und entsprechende Fehlermeldungen wurden ausgegeben.	
<b>Haupt-Neben-Akteur</b>	Benutzer	
<b>Auslöser</b>	Der Benutzer möchte Messdaten eingeben.	
<b>Standardfluss</b>	<b>Schritt</b>	<b>Aktion</b>
	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.
<b>Nebenfluss</b>	<b>Schritt</b>	<b>Aktion</b>
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

<b>Name</b>	Messungen importieren	
<b>Ziel</b>	Der Benutzer möchte experimentelle Daten importieren.	
<b>Einordnung</b>	Basisfunktion	
<b>Vorbedingung</b>	Die Software wird korrekt ausgeführt.	
<b>Nachbedingung</b>	Die ausgewählten Daten wurden geladen.	
<b>Nachbedingung im Fehlerfall</b>	Die Daten wurden nicht geändert und entsprechende Fehlermeldungen wurden ausgegeben.	
<b>Haupt-Neben-Akteur</b>	Benutzer	
<b>Auslöser</b>	Der Benutzer möchte Messdaten importieren.	
<b>Standardfluss</b>	<b>Schritt</b>	<b>Aktion</b>
	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.
<b>Nebenfluss</b>	<b>Schritt</b>	<b>Aktion</b>
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

<b>Name</b>	Randbedingungen eingeben	
<b>Ziel</b>	Der Benutzer möchte Randbedingungen vorgeben.	
<b>Einordnung</b>	Basisfunktion	
<b>Vorbedingung</b>	Die Software wird korrekt ausgeführt.	
<b>Nachbedingung</b>	Die Randbedingungen wurden vorgegeben und gespeichert.	
<b>Nachbedingung im Fehlerfall</b>	Die Randbedingungen wurden nicht geändert und entsprechende Fehlermeldungen wurden ausgegeben.	
<b>Haupt-Neben-Akteur</b>	Benutzer	
<b>Auslöser</b>	Der Benutzer möchte Randbedingungen vorgeben.	
<b>Standardfluss</b>	<b>Schritt</b>	<b>Aktion</b>
	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.
<b>Nebenfluss</b>	<b>Schritt</b>	<b>Aktion</b>
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

<b>Name</b>	Parameter optimieren	
<b>Ziel</b>	Der Benutzer möchte die Temperaturleitkoeffizienten optimal im Sinne kleinsten Fehlerquadrate an experimentelle Daten anpassen lassen.	
<b>Einordnung</b>	Hauptfunktion	
<b>Vorbedingung</b>	Die Software wird korrekt ausgeführt.	
<b>Nachbedingung</b>	Die Parameter wurden optimiert.	
<b>Nachbedingung im Fehlerfall</b>	Die Daten wurden nicht geändert und entsprechende Fehlermeldungen wurden ausgegeben.	
<b>Haupt-Neben-Akteur</b>	Benutzer	
<b>Auslöser</b>	Der Benutzer möchte die Temperaturleitkoeffizienten optimieren lassen.	
<b>Standardfluss</b>	<b>Schritt</b>	<b>Aktion</b>
	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

<b>Name</b>	Simulationsexperiment laden/speichern	
<b>Ziel</b>	Der Benutzer möchte ein Simulationsexperiment laden/spichern.	
<b>Einordnung</b>	Hauptfunktion	
<b>Vorbedingung</b>	Die Software wird korrekt ausgeführt.	
<b>Nachbedingung</b>	Das eingegebene Simulationsexperiment wurde geladen/gespeichert.	
<b>Nachbedingung im Fehlerfall</b>	Die Daten wurden nicht geändert und entsprechende Fehlermeldungen wurden ausgegeben.	
<b>Haupt-Neben-Akteur</b>	Benutzer	
<b>Auslöser</b>	Der Benutzer möchte ein Simulationsexperiment laden/speichern.	
<b>Standardfluss</b>	<b>Schritt</b>	<b>Aktion</b>
	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.
<b>Nebenfluss</b>	<b>Schritt</b>	<b>Aktion</b>
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

<b>Name</b>	Simulieren	
<b>Ziel</b>	Der Benutzer möchte simulieren.	
<b>Einordnung</b>	Hauptfunktion	
<b>Vorbedingung</b>	Die Software wird korrekt ausgeführt.	
<b>Nachbedingung</b>	Die Simulation wurde ausgeführt.	
<b>Nachbedingung im Fehlerfall</b>	Die Simulation wurden nicht ausgeführt und entsprechende Fehlermeldungen wurden ausgegeben.	
<b>Haupt-Neben-Akteur</b>	Benutzer	
<b>Auslöser</b>	Der Benutzer möchte die Simulation starten.	
<b>Standardfluss</b>	<b>Schritt</b>	<b>Aktion</b>
	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

<b>Name</b>	Temperaturleitkoeffizienten eingeben	
<b>Ziel</b>	Der Benutzer möchte Temperaturleitkoeffizienten eingeben.	
<b>Einordnung</b>	Basisfunktion	
<b>Vorbedingung</b>	Die Software wird korrekt ausgeführt.	
<b>Nachbedingung</b>	Die Temperaturleitkoeffizienten wurden eingegeben und gespeichert.	
<b>Nachbedingung im Fehlerfall</b>	Die Temperaturleitkoeffizienten wurden nicht geändert und entsprechende Fehlermeldungen wurden ausgegeben.	
<b>Haupt-Neben-Akteur</b>	Benutzer	
<b>Auslöser</b>	Der Benutzer möchte Temperaturleitkoeffizienten eingeben.	
<b>Standardfluss</b>	<b>Schritt</b>	<b>Aktion</b>
	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.
<b>Nebenfluss</b>	<b>Schritt</b>	<b>Aktion</b>
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

<b>Name</b>	Video abspielen	
<b>Ziel</b>	Der Benutzer möchte die zeitliche Entwicklung der Temperaturverteilung untersuchen.	
<b>Einordnung</b>	Hauptfunktion	
<b>Vorbedingung</b>	Die Software wird korrekt ausgeführt und es wurde eine Simulation erfolgreich durchgeführt.	
<b>Nachbedingung</b>	Das Video wird abgespielt.	
<b>Nachbedingung im Fehlerfall</b>	Das Video wurde nicht abgespielt und entsprechende Fehlermeldungen wurden ausgegeben.	
<b>Haupt-Neben-Akteur</b>	Benutzer	
<b>Auslöser</b>	Der Benutzer möchte die zeitliche Entwicklung der Temperaturverteilung untersuchen.	
<b>Standardfluss</b>	<b>Schritt</b>	<b>Aktion</b>
	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

<b>Name</b>	Wärmequellen eingeben	
<b>Ziel</b>	Der Benutzer möchte Wärmequellen eingeben.	
<b>Einordnung</b>	Basisfunktion	
<b>Vorbedingung</b>	Die Software wird korrekt ausgeführt.	
<b>Nachbedingung</b>	Die Wärmequellen wurden eingegeben und gespeichert.	
<b>Nachbedingung im Fehlerfall</b>	Die Wärmequellen wurden nicht geändert und entsprechende Fehlermeldungen wurden ausgegeben.	
<b>Haupt-Neben-Akteur</b>	Benutzer	
<b>Auslöser</b>	Der Benutzer möchte Wärmequellen eingeben.	
<b>Standardfluss</b>	<b>Schritt</b>	<b>Aktion</b>
	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.
<b>Nebenfluss</b>	<b>Schritt</b>	<b>Aktion</b>
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

<b>Name</b>	Zustand anzeigen	
<b>Ziel</b>	Der Benutzer möchte ein Zustand anzeigen lassen.	
<b>Einordnung</b>	Hauptfunktion	
<b>Vorbedingung</b>	Die Software wird korrekt ausgeführt und es wurde eine Simulation erfolgreich durchgeführt.	
<b>Nachbedingung</b>	Der Zustand wird angezeigt.	
<b>Nachbedingung im Fehlerfall</b>	Der Zustand wurde nicht angezeigt und entsprechende Fehlermeldungen wurden ausgegeben.	
<b>Haupt-Neben-Akteur</b>	Benutzer	
<b>Auslöser</b>	Der Benutzer möchte ein Zustand anzeigen lassen.	
<b>Standardfluss</b>	<b>Schritt</b>	<b>Aktion</b>
	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. 2.3 - 2.14) zeigen die Aktivitätsdiagramme der Anwendungsfälle.

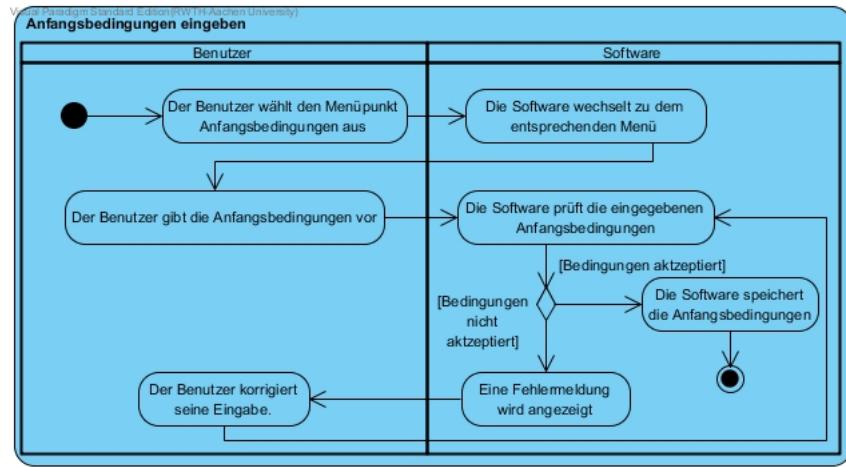


Abbildung 2.3: Aktivitätsdiagramm Use Case Anfangsbedingungen eingeben

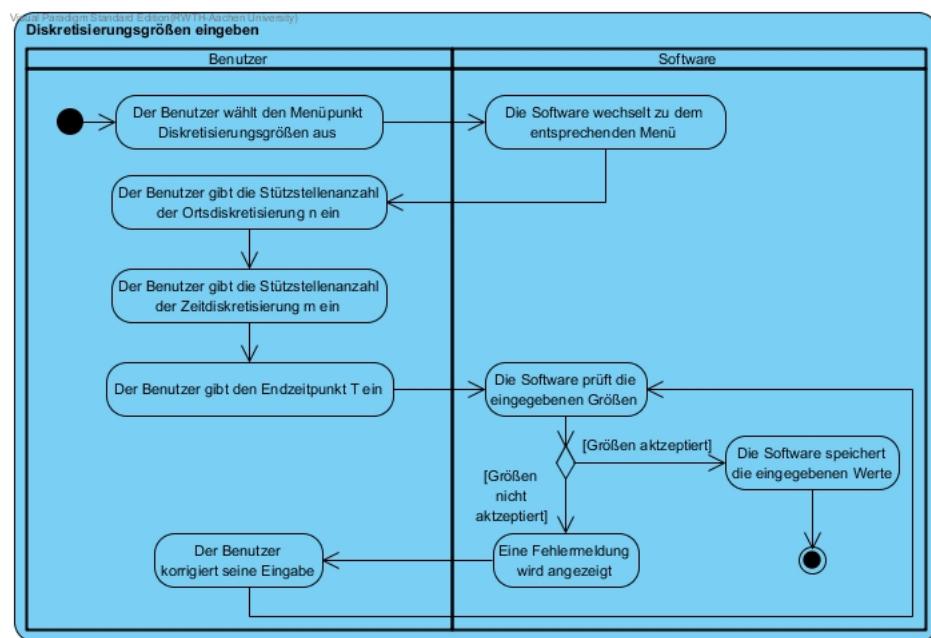


Abbildung 2.4: Aktivitätsdiagramm Use Case Diskretisierungsgrößen eingeben

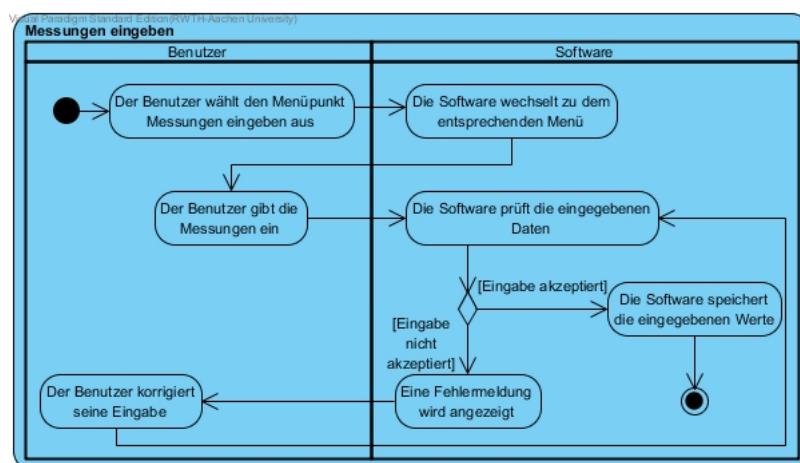


Abbildung 2.5: Aktivitätsdiagramm Use Case Messungen eingeben

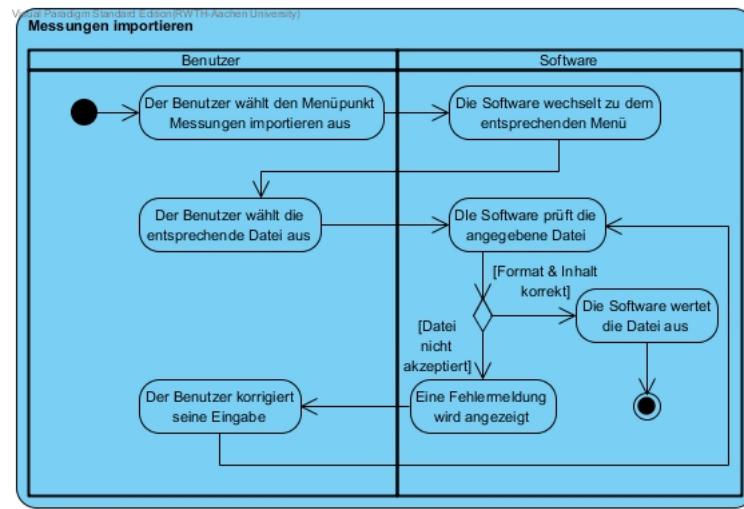


Abbildung 2.6: Aktivitätsdiagramm Use Case Messungen importieren

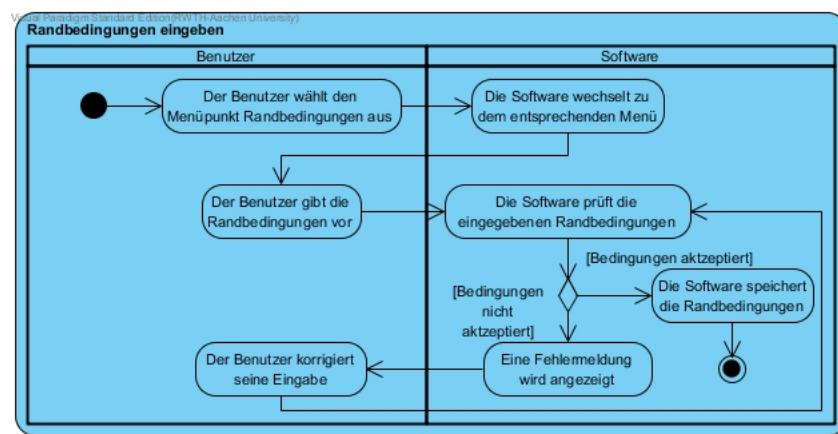


Abbildung 2.7: Aktivitätsdiagramm Use Case Randbedingungen eingeben

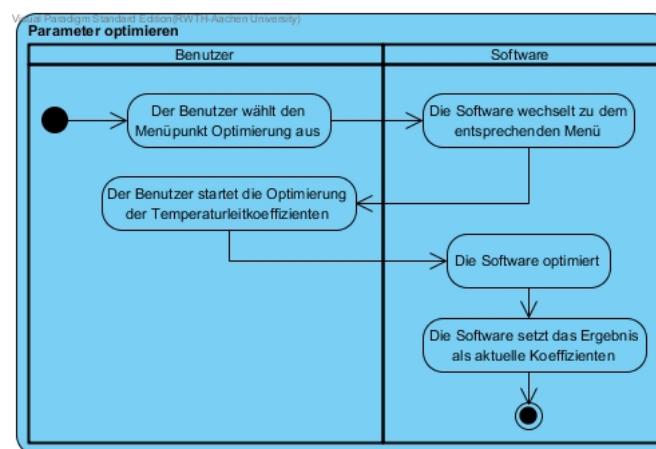


Abbildung 2.8: Aktivitätsdiagramm Use Case Parameter optimieren

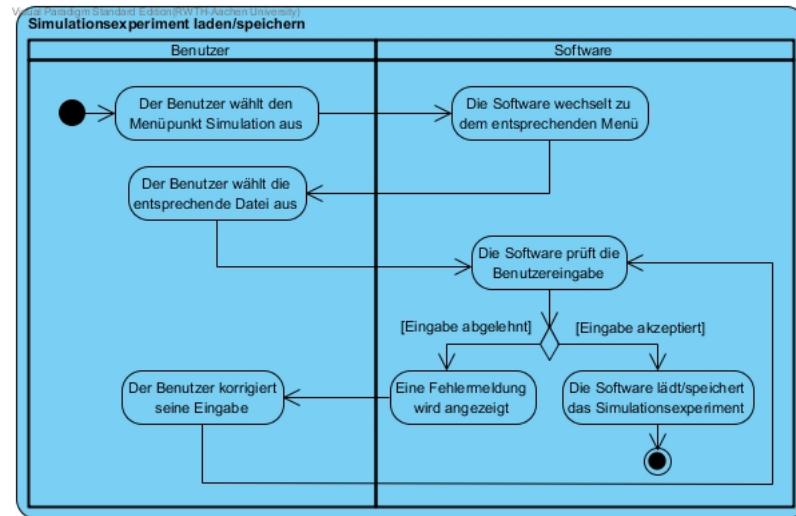


Abbildung 2.9: Aktivitätsdiagramm Use Case Simulationsexperiment laden/speichern

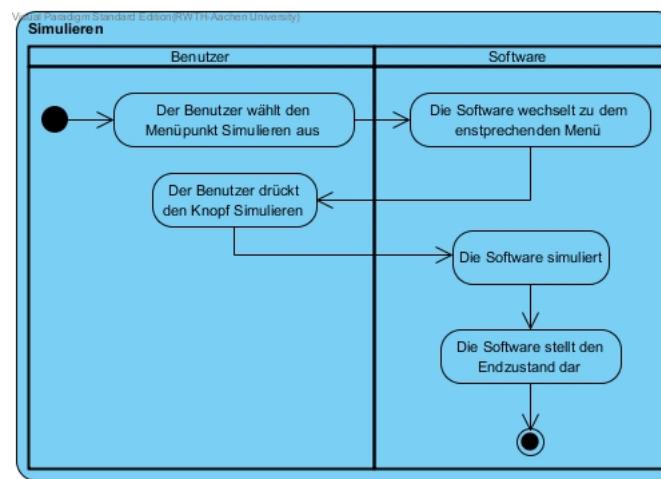


Abbildung 2.10: Aktivitätsdiagramm Use Case Simulieren

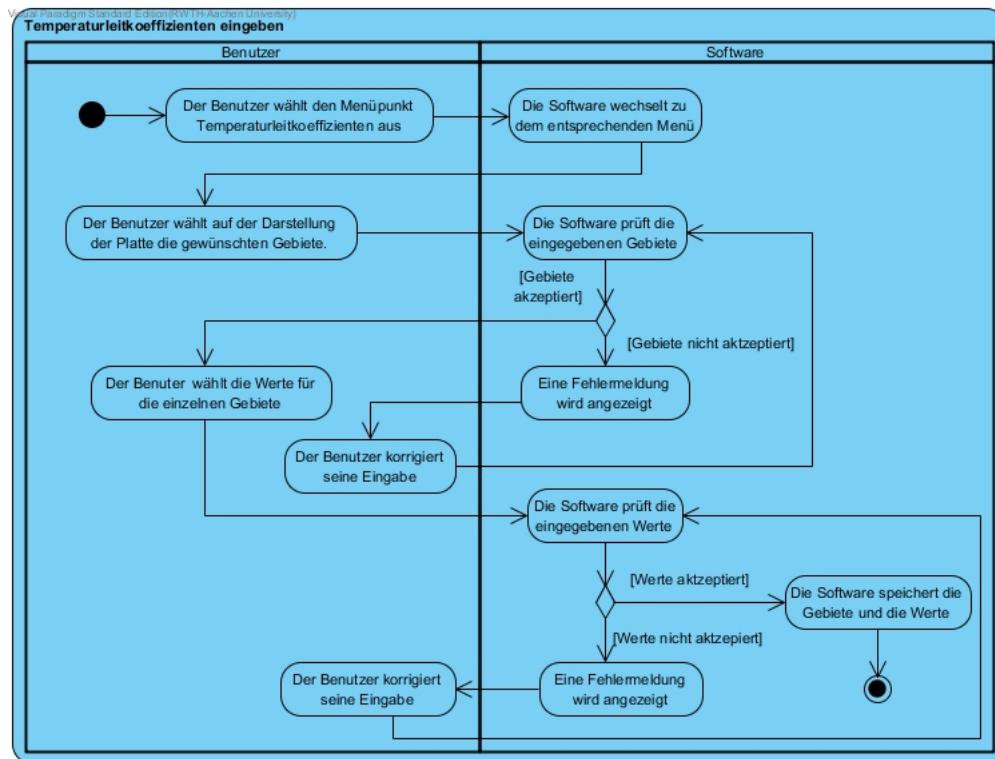


Abbildung 2.11: Aktivitätsdiagramm Use Case Temperaturleitkoeffizienten eingeben

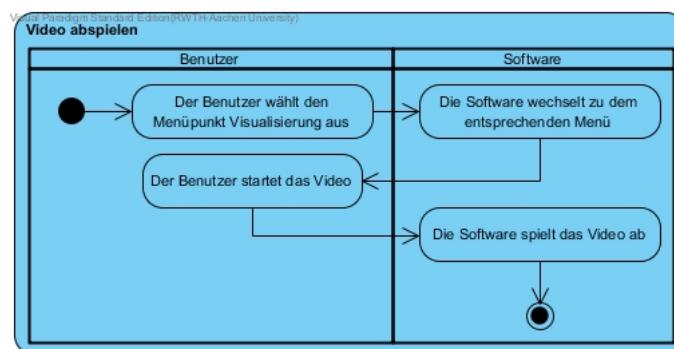


Abbildung 2.12: Aktivitätsdiagramm Use Case Video abspielen

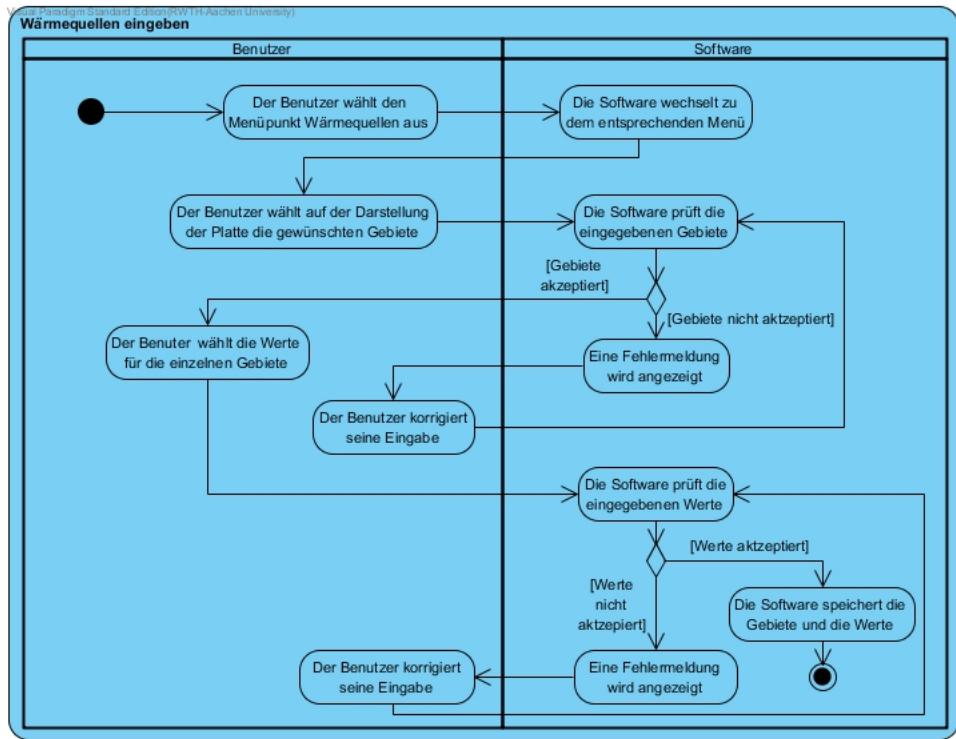


Abbildung 2.13: Aktivitätsdiagramm Use Case Wärmequellen eingeben

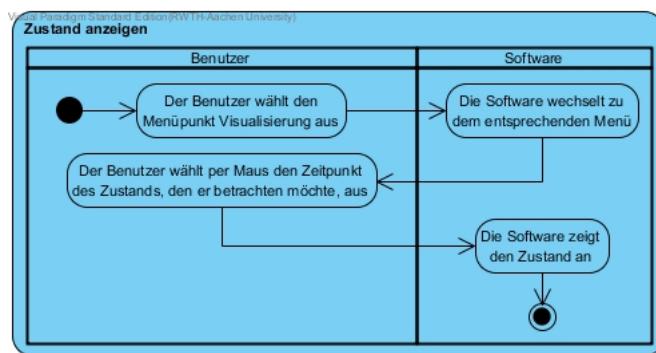


Abbildung 2.14: Aktivitätsdiagramm Use Case Zustand anzeigen

#### 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 2.15 zeigt das Begriffsnetz.

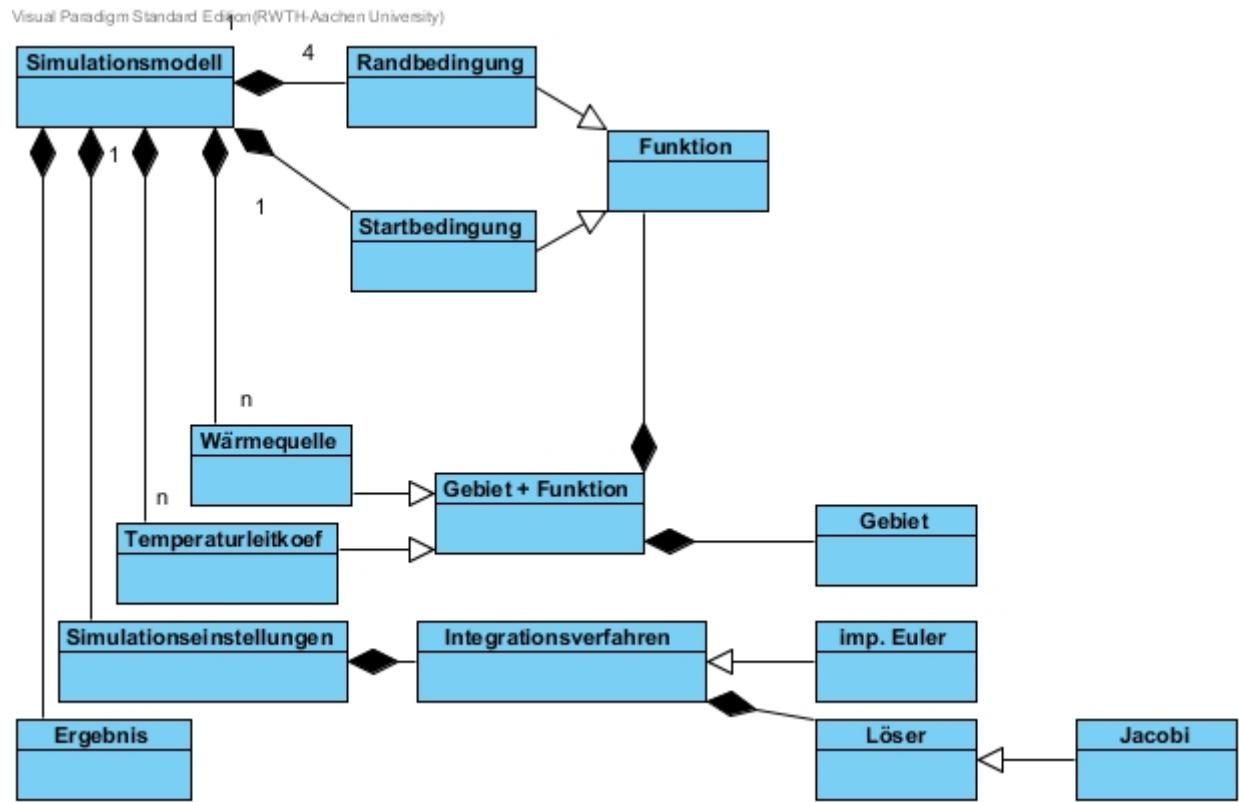


Abbildung 2.15: Begriffsnetz

# Kapitel 3

## Entwurf

### 3.1 Pakete

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

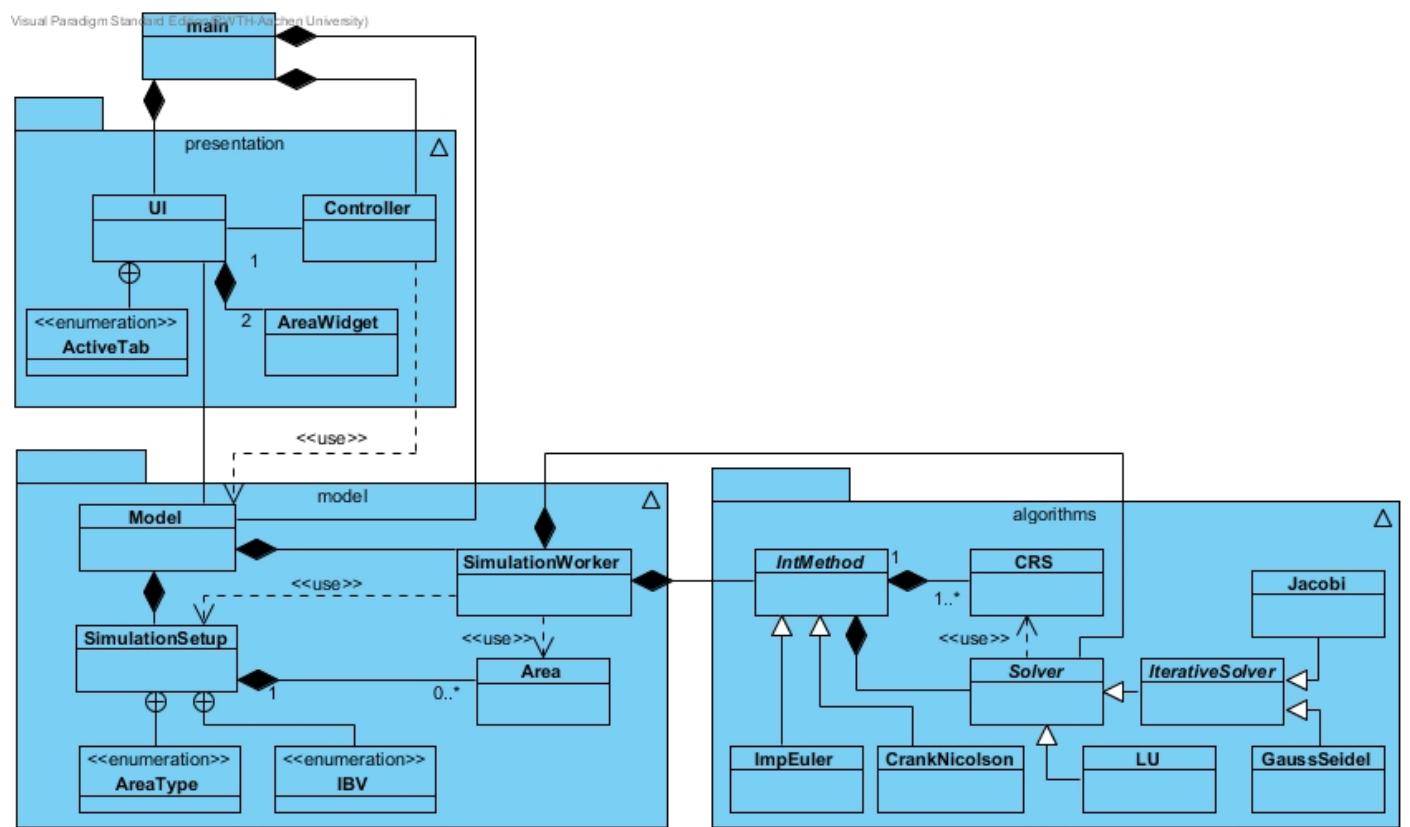


Abbildung 3.1: Paketstruktur

## 3.2 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 keine reinen Getter/Setter-Funktionen oder Funktionen die lediglich Algorithmen implementieren.

### 3.2.1 Paket algorithms

Das Klassendiagramm in Abbildung 3.2 zeigt alle im Paket *algorithms* enthaltene Klassen und deren Attribute und Funktionen.

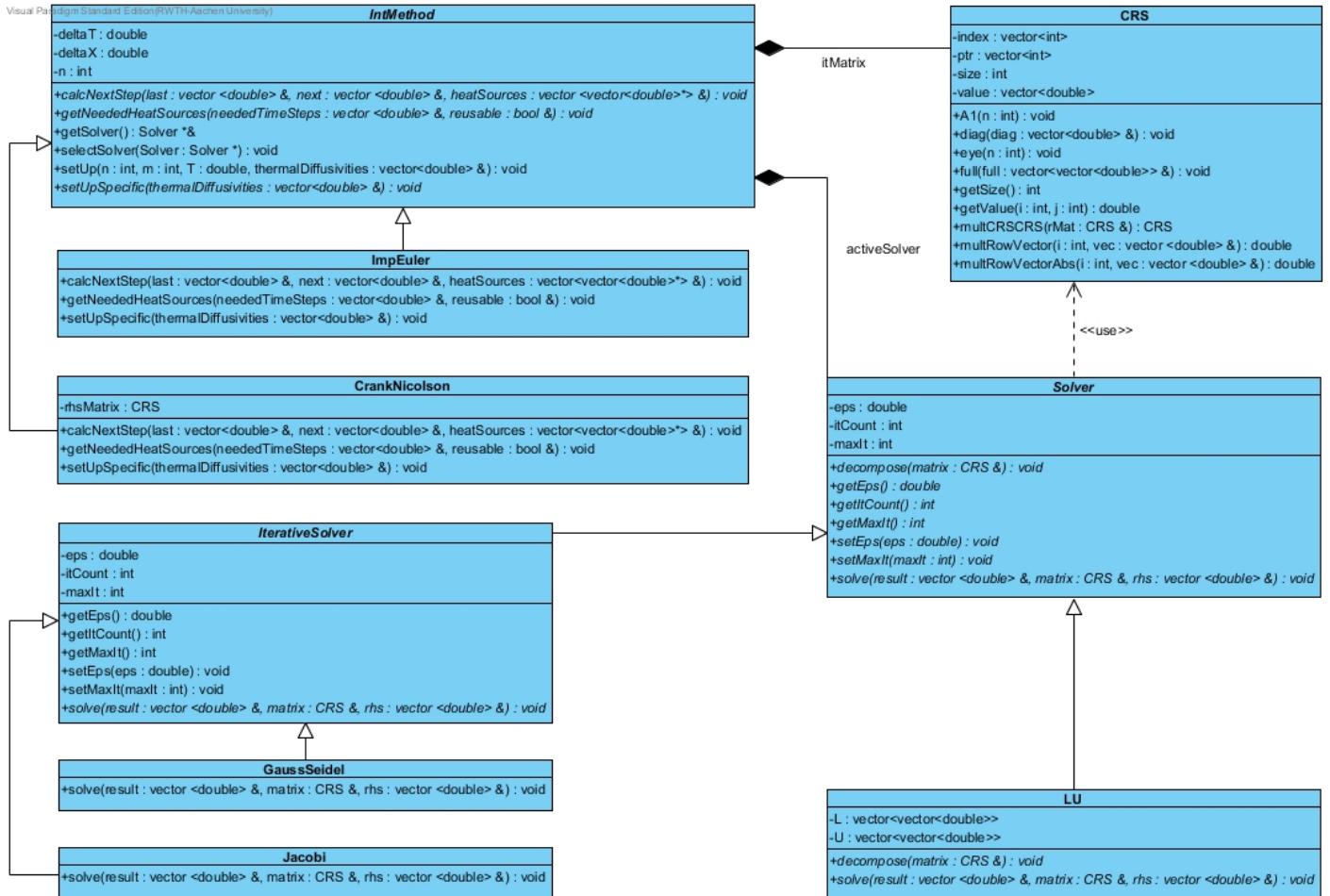


Abbildung 3.2: Klassendiagramm algorithms

### 3.2.1.1 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.

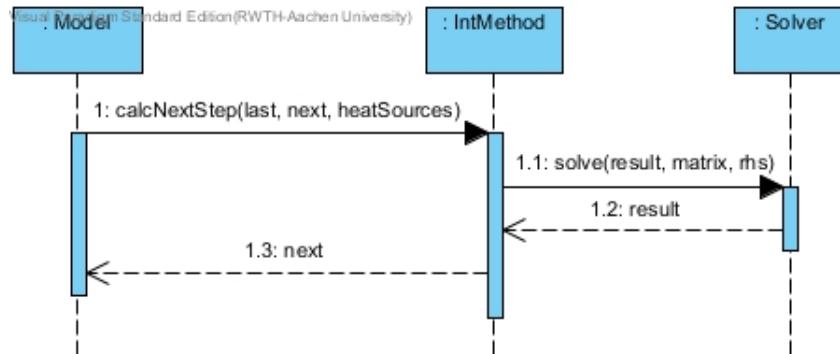


Abbildung 3.3: Sequenzdiagramm IntMethod::calcNextStep

#### setUpSpecific

Das Sequenzdiagramm für *setUpSpecific* ist in Abbildung 3.4 dargestellt. *setUpSpecific* trifft für die gewählte Integrationsmethode spezifische Vorbereitungen.

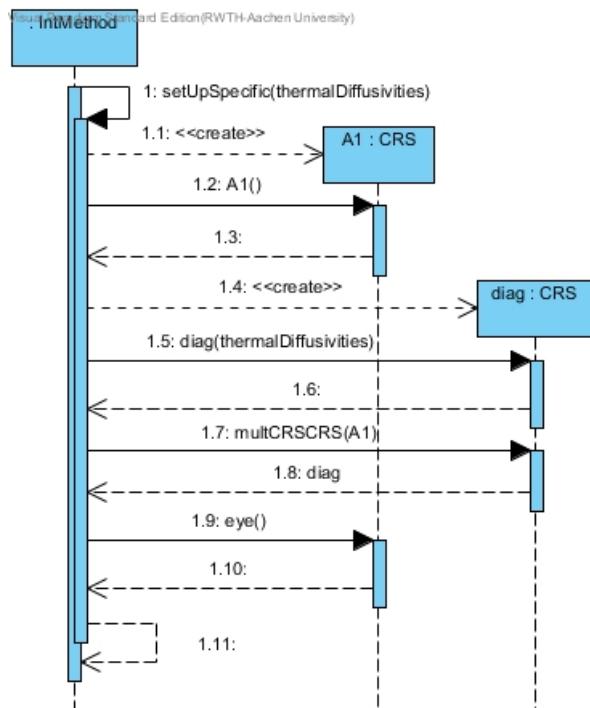


Abbildung 3.4: Sequenzdiagramm setUpSpecific

### 3.2.2 Paket model

Das Klassendiagramm in Abbildung 3.5 zeigt alle im Paket *model* enthaltene Klassen und deren Attribute und Funktionen.

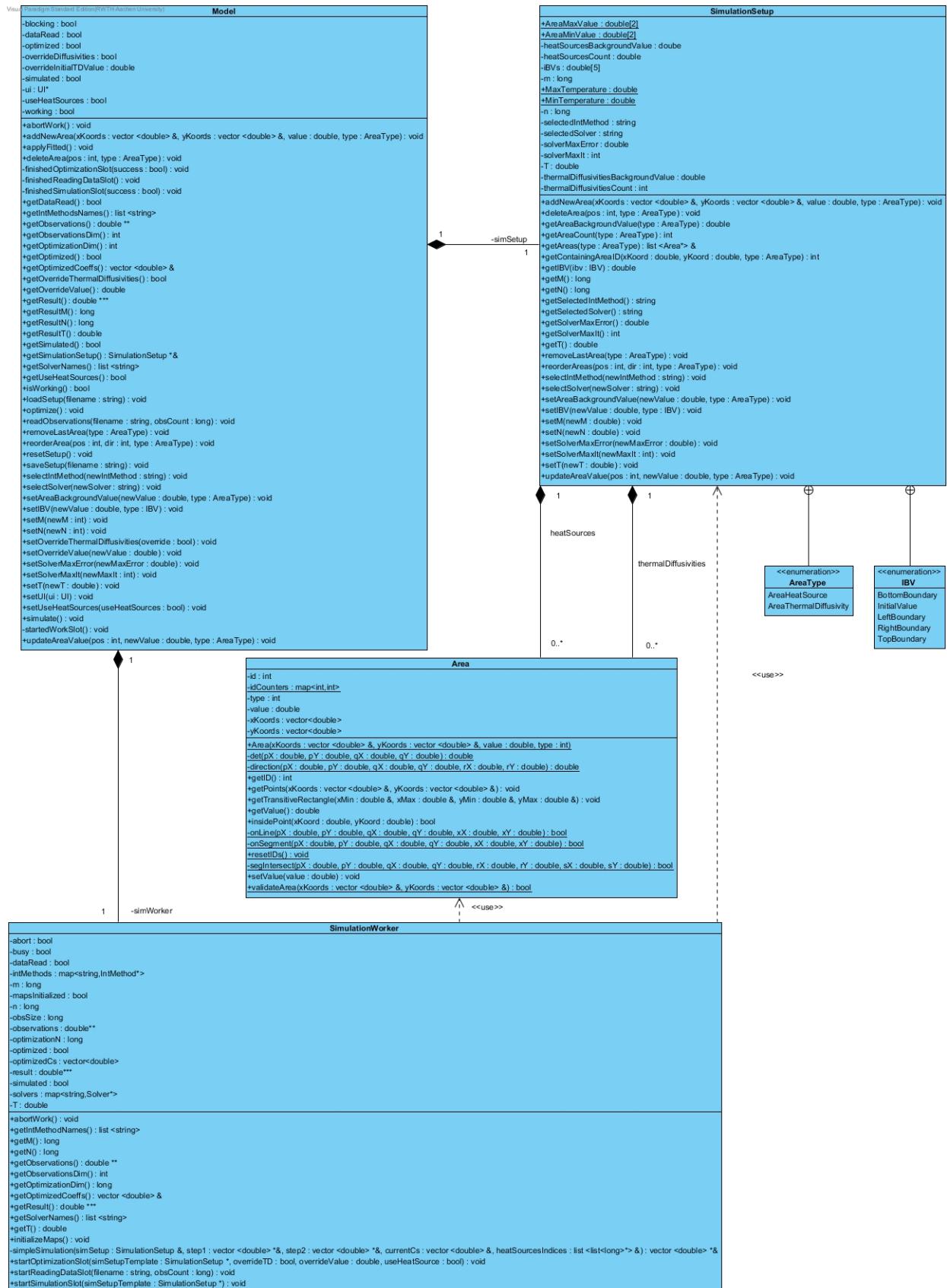


Abbildung 3.5: Klassendiagramm Model

### 3.2.2.1 Model

#### abortWork

Das Sequenzdiagramm für *abortWork* ist in Abbildung 3.6 dargestellt. *abortWork* bricht eine laufende Simulation oder Optimierung ab.

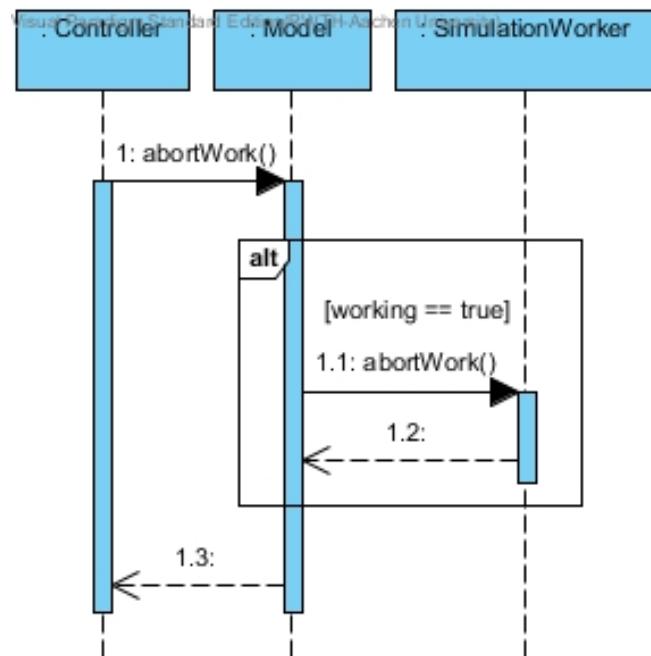


Abbildung 3.6: Sequenzdiagramm Model::abortWork

#### addNewArea

Das Sequenzdiagramm für *addNewArea* ist in Abbildung 3.7 dargestellt. *addNewArea* fügt ein neues Gebiet vom übergebenen Typ hinzu.

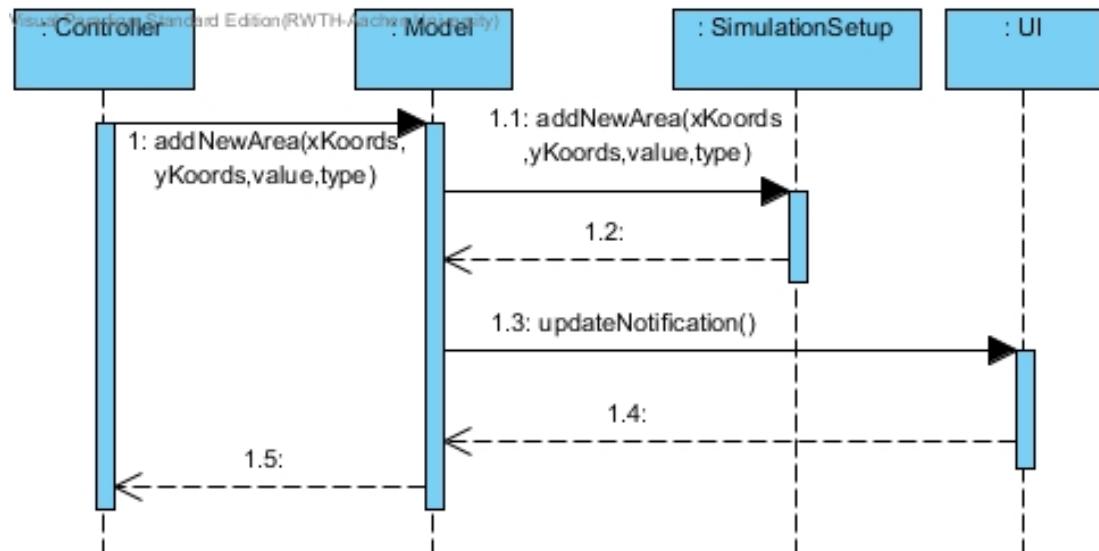


Abbildung 3.7: Sequenzdiagramm Model::addNewArea

#### deleteArea

Das Sequenzdiagramm für *deleteArea* ist in Abbildung 3.8 dargestellt. *deleteArea* löscht das Gebiet vom übergebenen Typ an der übergebenen Position.

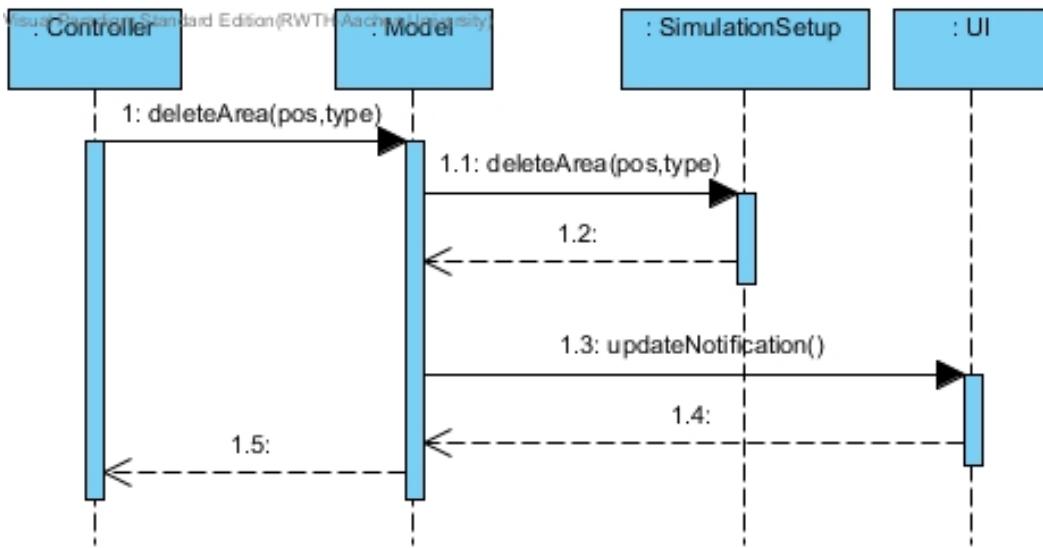


Abbildung 3.8: Sequenzdiagramm Model::deleteArea

### `finishedOptimizationSlot`

Das Sequenzdiagramm für `finishedOptimizationSlot` ist in Abbildung 3.9 dargestellt. `finishedOptimizationSlot` informiert das UI über das Ende einer Optimierung.

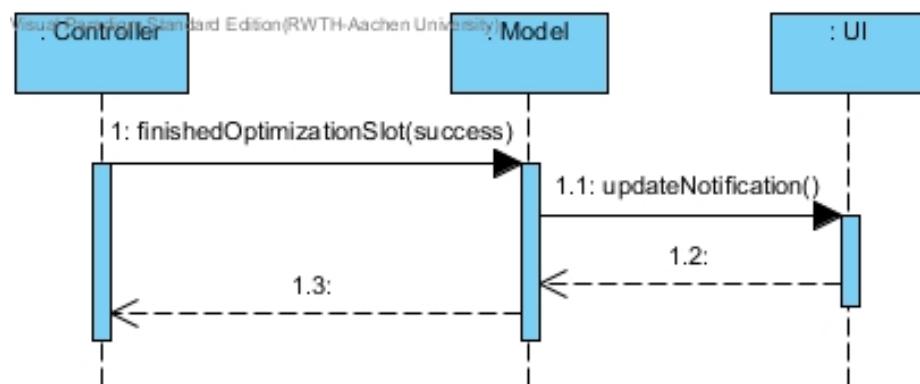


Abbildung 3.9: Sequenzdiagramm Model::finishedOptimizationSlot

### `finishedReadingDataSlot`

Das Sequenzdiagramm für `finishedReadingDataSlot` ist in Abbildung 3.10 dargestellt. `finishedReadingDataSlot` informiert das UI über das Ende des Einlesen von Messdaten.

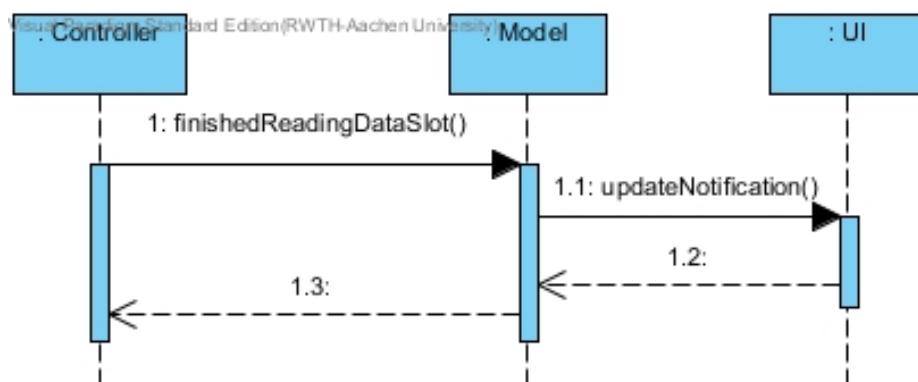


Abbildung 3.10: Sequenzdiagramm Model::finishedReadingDataSlot

## finishedSimulationSlot

Das Sequenzdiagramm für *finishedSimulationSlot* ist in Abbildung 3.11 dargestellt. *finishedSimulationSlot* informiert das UI über das Ende einer Simulation.

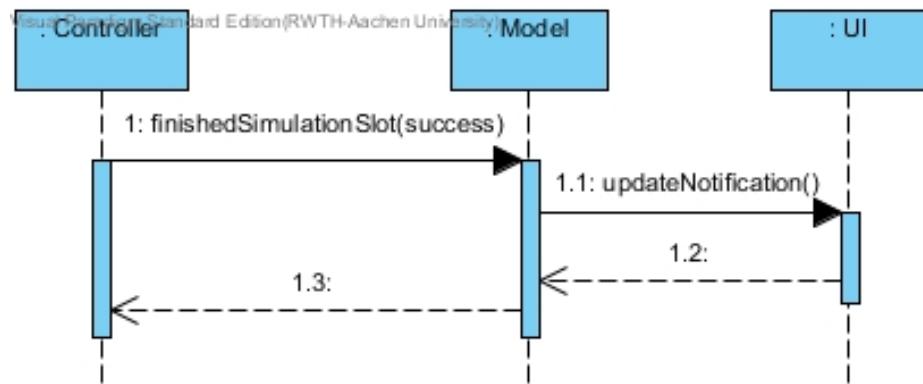


Abbildung 3.11: Sequenzdiagramm Model::finishedSimulationSlot

## getObservations

Das Sequenzdiagramm für *getObservations* ist in Abbildung 3.12 dargestellt. *getObservations* gibt eingelesene Messdaten zurück, falls bereits welche eingelesen wurden und gerade keine Simulation/Optimierung durchgeführt wird.

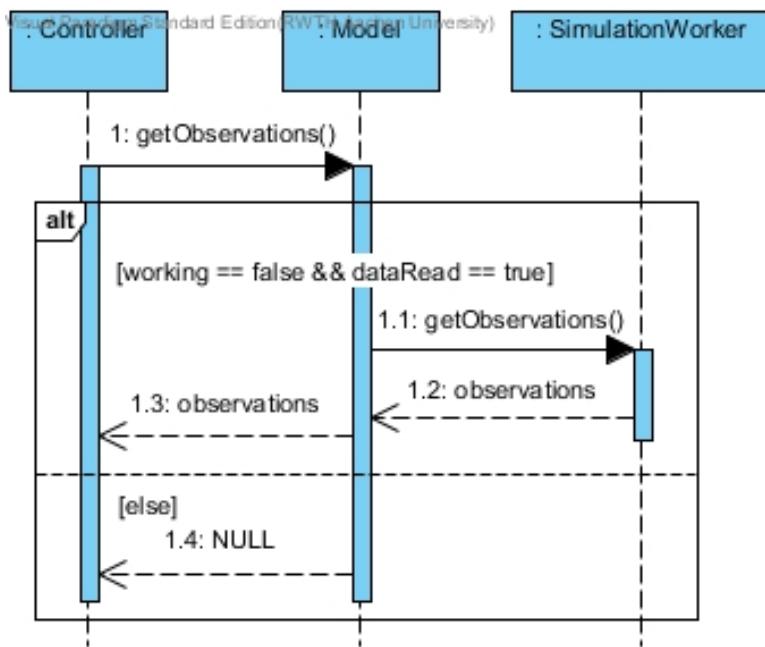


Abbildung 3.12: Sequenzdiagramm Model::getObservations

## getResult

Das Sequenzdiagramm für *getResult* ist in Abbildung 3.13 dargestellt. *getResult* gibt das Ergebnis einer Simulation zurück, falls bereits eine durchgeführt wurde und gerade keine Simulation/Optimierung durchgeführt wird.

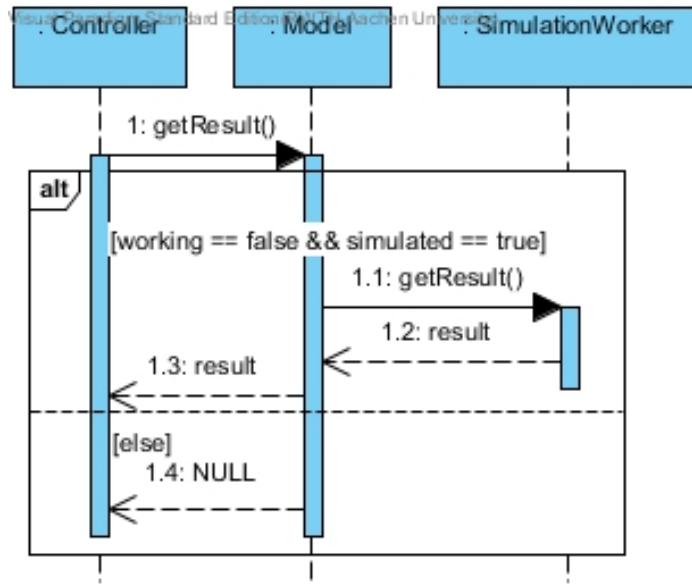


Abbildung 3.13: Sequenzdiagramm Model::getResult

### getResultM

Das Sequenzdiagramm für `getResultM` ist in Abbildung 3.14 dargestellt. `getResult` gibt die Zeitdiskretisierungsgröße der letzten Simulation zurück, falls bereits eine durchgeführt wurde und gerade keine Simulation/Optimierung durchgeführt wird.

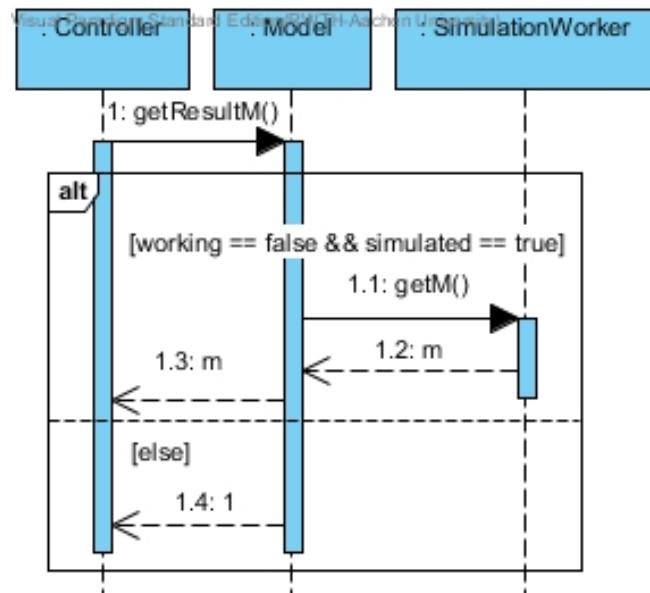


Abbildung 3.14: Sequenzdiagramm Model::getResultM

### getResultN

Das Sequenzdiagramm für `getResultN` ist in Abbildung 3.15 dargestellt. `getResultN` gibt die Ortsdiskretisierungsgröße der letzten Simulation zurück, falls bereits eine durchgeführt wurde und gerade keine Simulation/Optimierung durchgeführt wird.

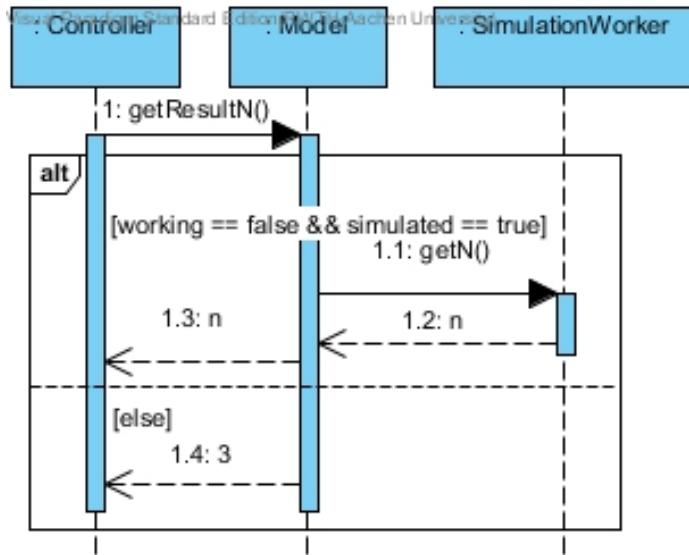


Abbildung 3.15: Sequenzdiagramm Model::getResultN

### getResultT

Das Sequenzdiagramm für `getResultT` ist in Abbildung 3.16 dargestellt. `getResultT` gibt den Endzeitpunkt der letzten Simulation zurück, falls bereits eine durchgeführt wurde und gerade keine Simulation/Optimierung durchgeführt wird.

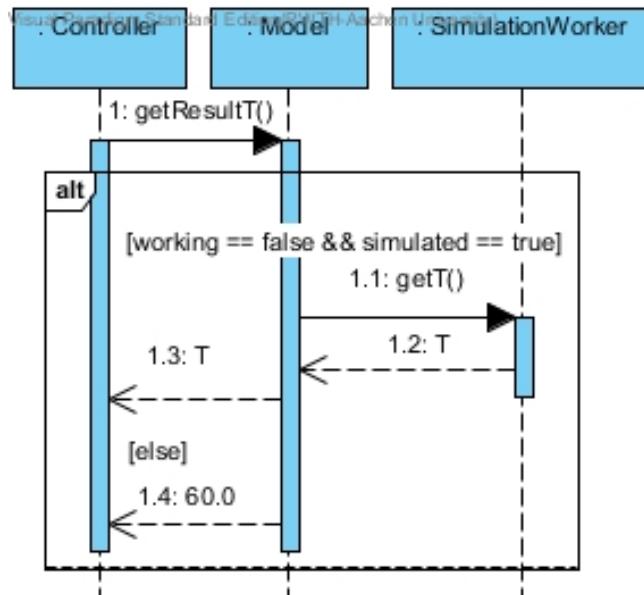


Abbildung 3.16: Sequenzdiagramm Model::getResultT

### loadSetup

Das Sequenzdiagramm für `loadSetup` ist in Abbildung 3.17 dargestellt. `loadSetup` lädt gespeicherte Simulationseinstellungen aus einer Datei ein.

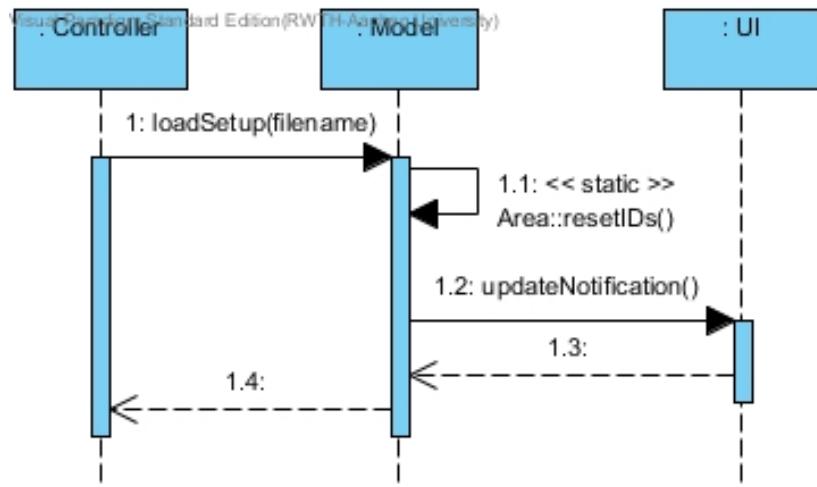


Abbildung 3.17: Sequenzdiagramm Model::loadSetup

### optimize

Das Sequenzdiagramm für *optimize* ist in Abbildung 3.18 dargestellt. *optimize* startet eine Optimierung der Temperaturleitkoeffizienten.

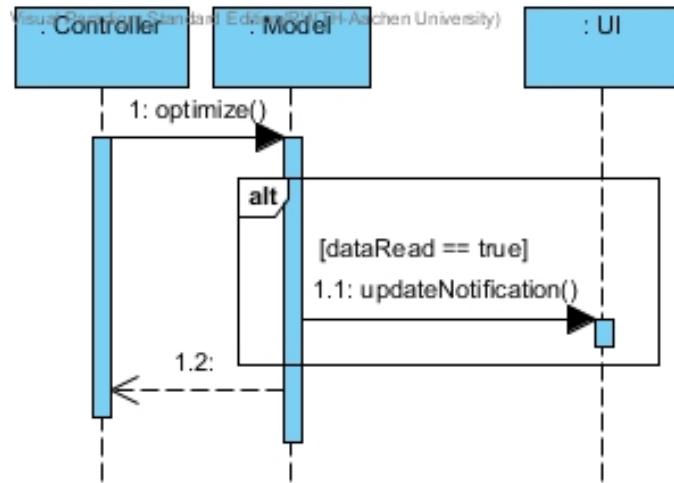


Abbildung 3.18: Sequenzdiagramm Model::optimize

### readObservations

Das Sequenzdiagramm für *readObservations* ist in Abbildung 3.19 dargestellt. *readObservations* startet das Einlesen von Messdaten für eine Optimierung.

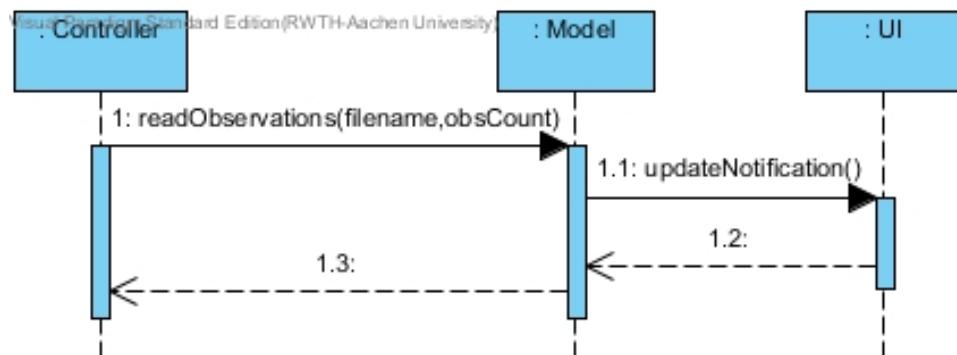


Abbildung 3.19: Sequenzdiagramm Model::readObservations

## removeLastArea

Das Sequenzdiagramm für *removeLastArea* ist in Abbildung 3.20 dargestellt. *removeLastArea* entfernt das letzte hinzugefügte Gebiet vom übergebenen Typ.

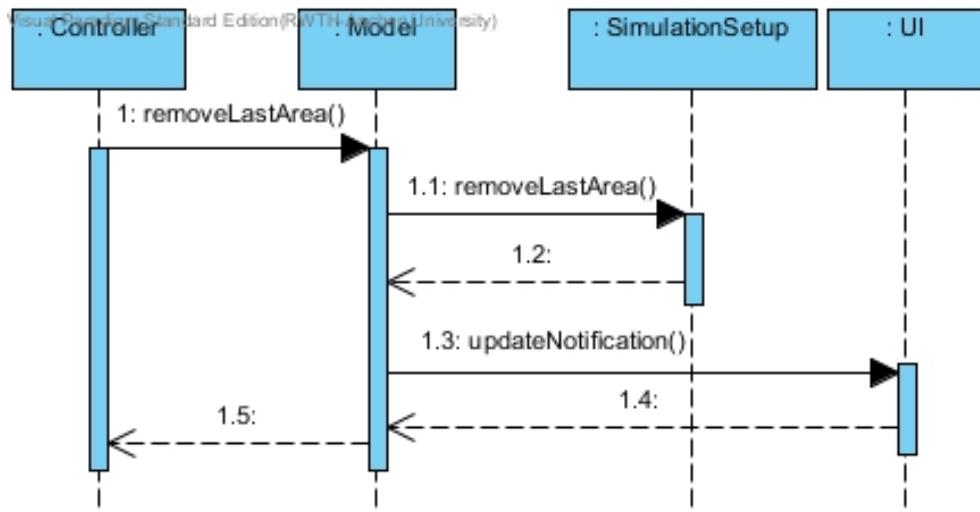


Abbildung 3.20: Sequenzdiagramm Model::removeLastArea

## reorderArea

Das Sequenzdiagramm für *reorderArea* ist in Abbildung 3.21 dargestellt. *reorderArea* ändert die Reihenfolge der Gebiete vom übergebenen Typ.

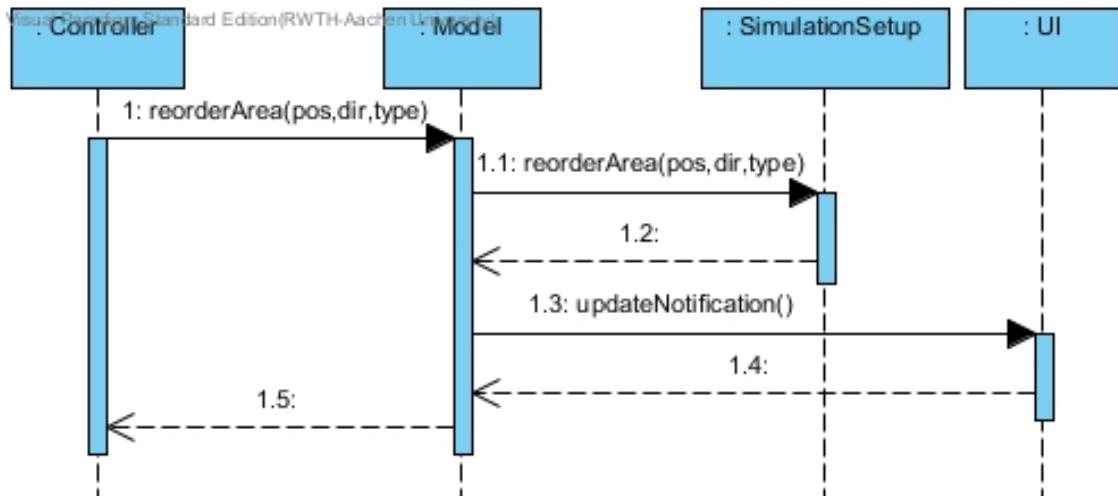


Abbildung 3.21: Sequenzdiagramm Model::reorderArea

## resetSetup

Das Sequenzdiagramm für *resetSetup* ist in Abbildung 3.22 dargestellt. *resetSetup* setzt die Simulationseinstellungen auf die Standardwerte zurück.

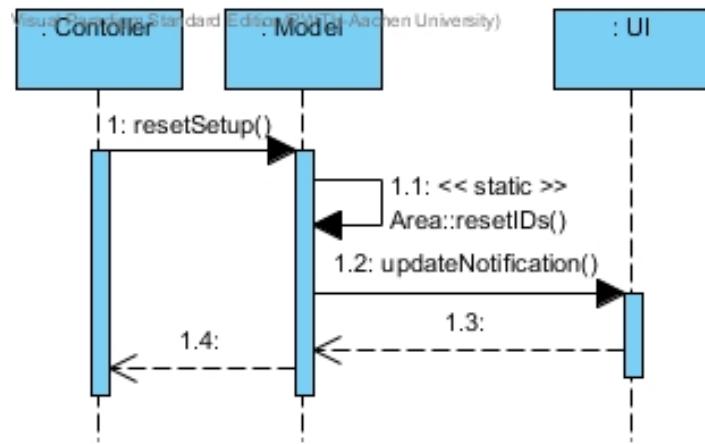


Abbildung 3.22: Sequenzdiagramm Model::resetSetup

### `selectIntMethod`

Das Sequenzdiagramm für `selectIntMethod` ist in Abbildung 3.23 dargestellt. `selectIntMethod` setzt die Integrationsmethode für die Simulation.

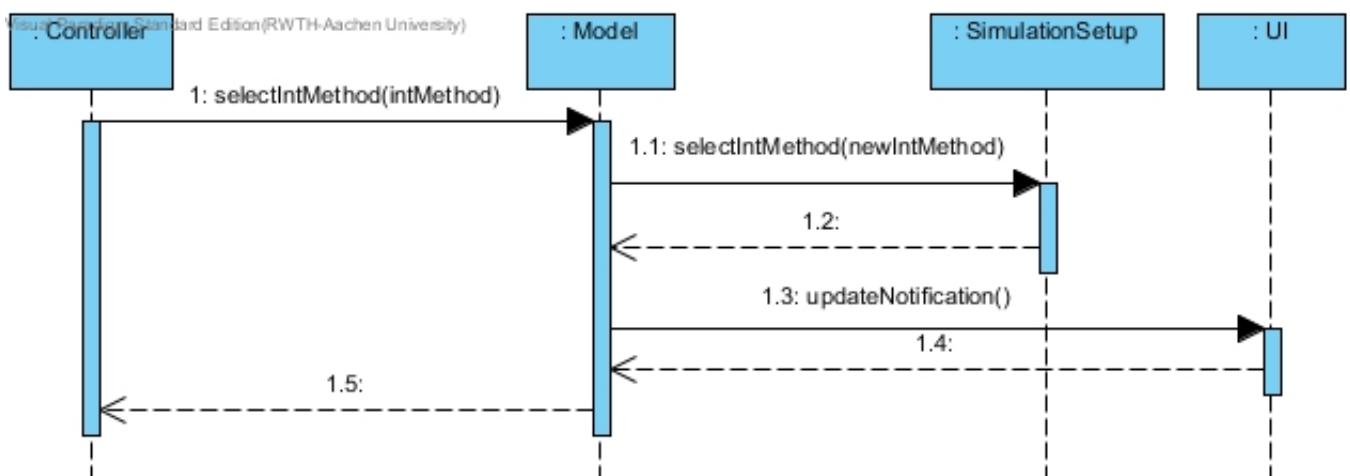


Abbildung 3.23: Sequenzdiagramm Model::selectIntMethod

### `selectSolver`

Das Sequenzdiagramm für `selectSolver` ist in Abbildung 3.24 dargestellt. `selectSolver` setzt den LGS Löser für die Simulation.

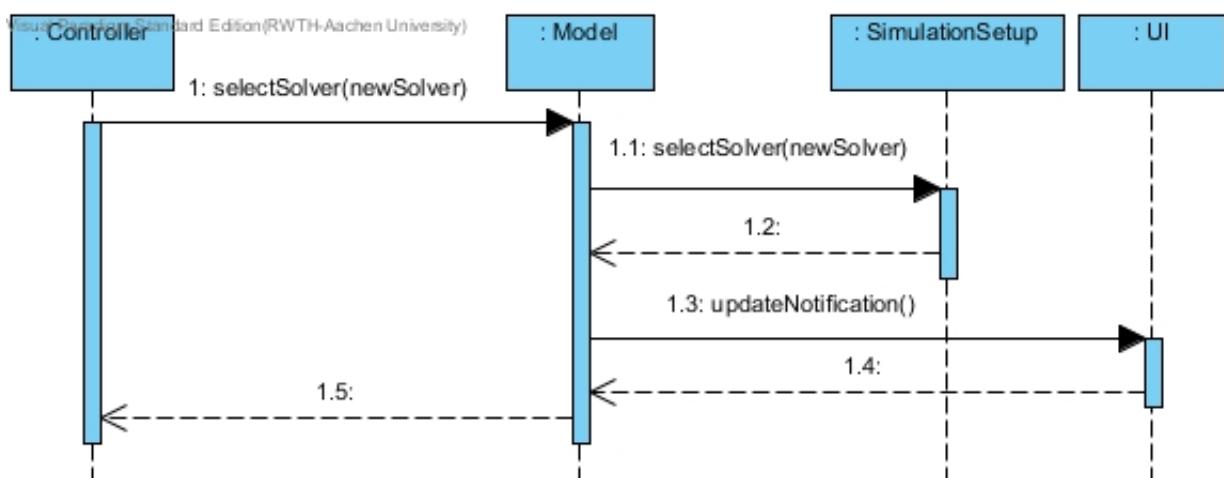


Abbildung 3.24: Sequenzdiagramm Model::selectSolver

## **setAreaBackgroundValue**

Das Sequenzdiagramm für *setAreaBackgroundValue* ist in Abbildung 3.25 dargestellt. *setAreaBackgroundValue* setzt den Hintergrundwert für die Gebiete des übergebenen Typs.

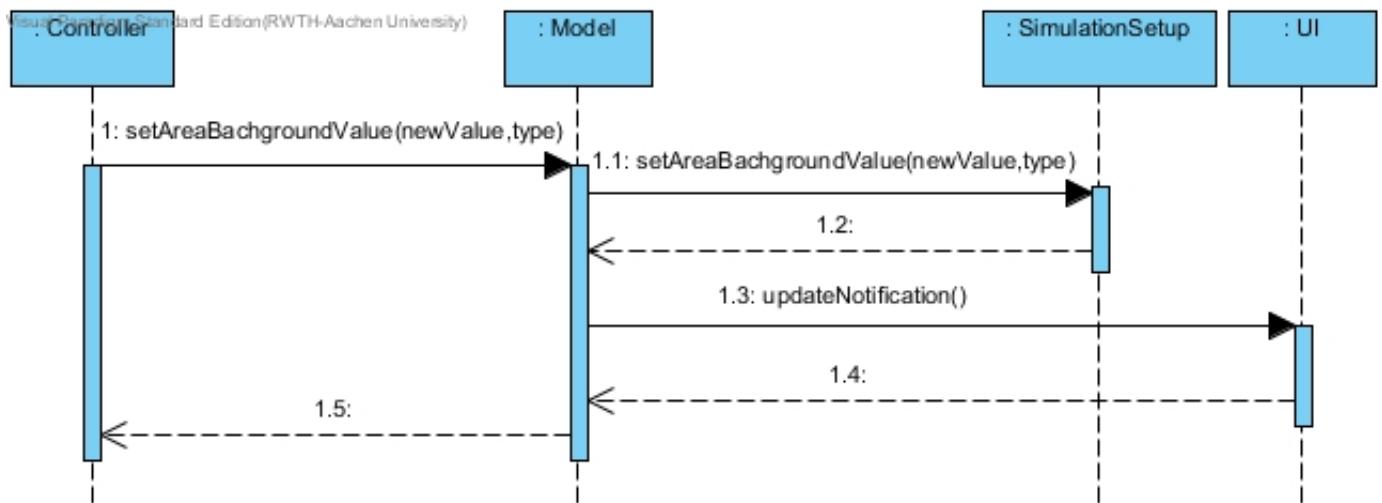


Abbildung 3.25: Sequenzdiagramm Model::setAreaBackgroundValue

## **setIBV**

Das Sequenzdiagramm für *setIBV* ist in Abbildung 3.26 dargestellt. *setIBV* setzt einen der Randwerte oder den Anfangswert für die Simulation.

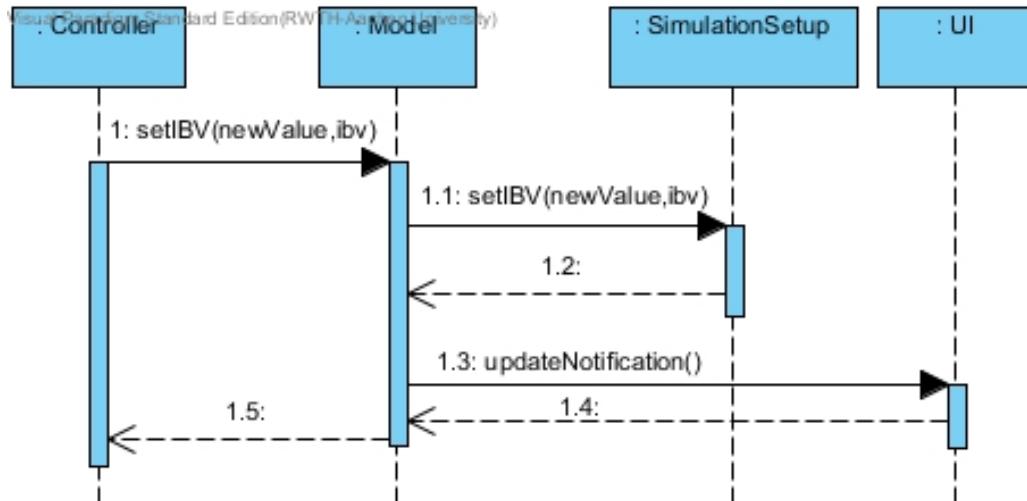


Abbildung 3.26: Sequenzdiagramm Model::setIBV

## **setM**

Das Sequenzdiagramm für *setM* ist in Abbildung 3.27 dargestellt. *setM* setzt die Zeitdiskretisierungsgröße für die Simulation.

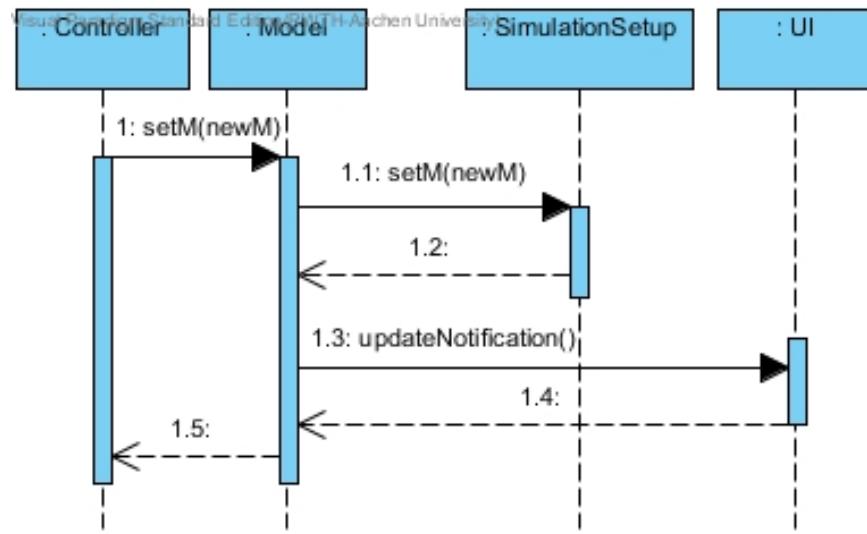


Abbildung 3.27: Sequenzdiagramm Model::setM

### setN

Das Sequenzdiagramm für *setN* ist in Abbildung 3.28 dargestellt. *setN* setzt die Ortsdiskretisierungsgröße für die Simulation.

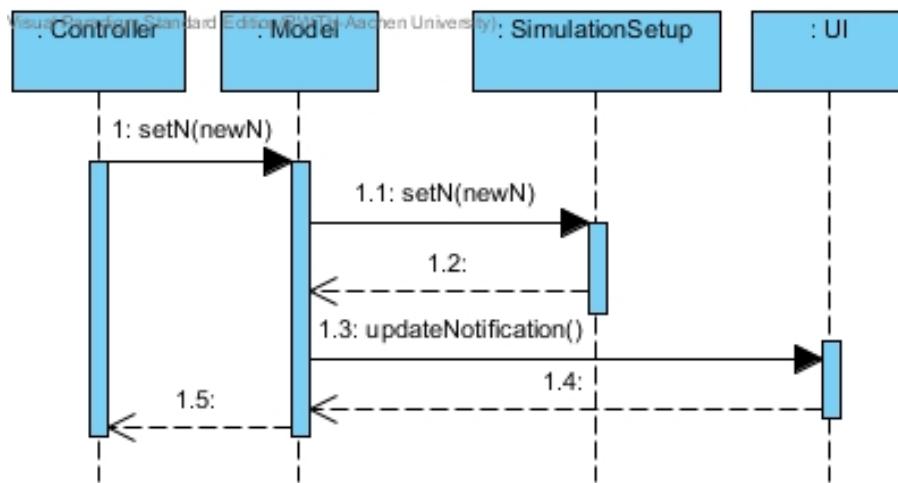


Abbildung 3.28: Sequenzdiagramm Model::setN

### setOverrideThermalDiffusivities

Das Sequenzdiagramm für *setOverrideThermalDiffusivities* ist in Abbildung 3.29 dargestellt. *setOverrideThermalDiffusivities* aktiviert die Nutzung des manuellen Anfangswertes für die Optimierung.

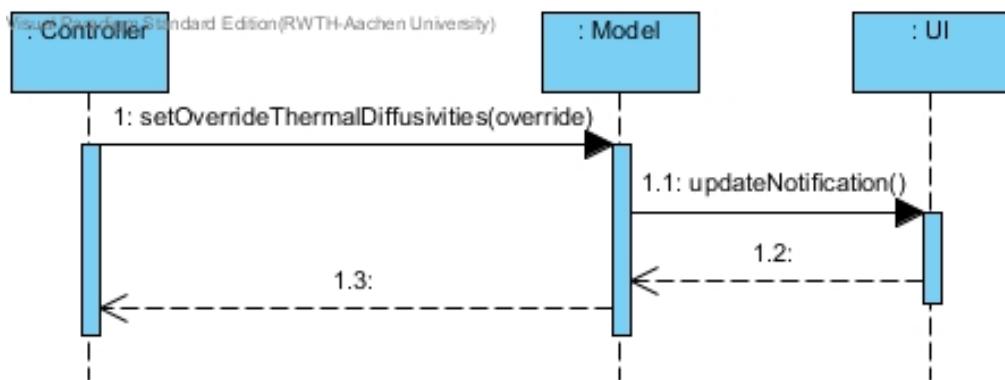


Abbildung 3.29: Sequenzdiagramm Model::setOverrideThermalDiffusivities

### **setOverrideValue**

Das Sequenzdiagramm für *setOverrideValue* ist in Abbildung 3.30 dargestellt. *setOverrideValue* setzt den manuellen Anfangswert für die Optimierung.

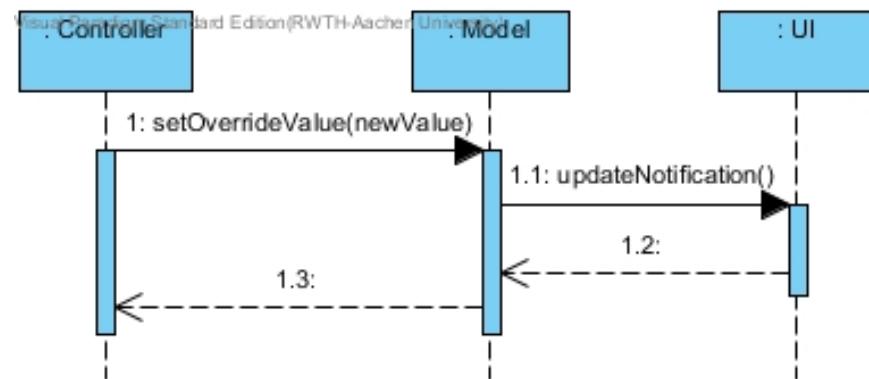


Abbildung 3.30: Sequenzdiagramm Model::setOverrideValue

### **setSolverMaxError**

Das Sequenzdiagramm für *setSolverMaxError* ist in Abbildung 3.31 dargestellt. *setSolverMaxError* setzt die relative Fehlertoleranz des LGS Lözers.

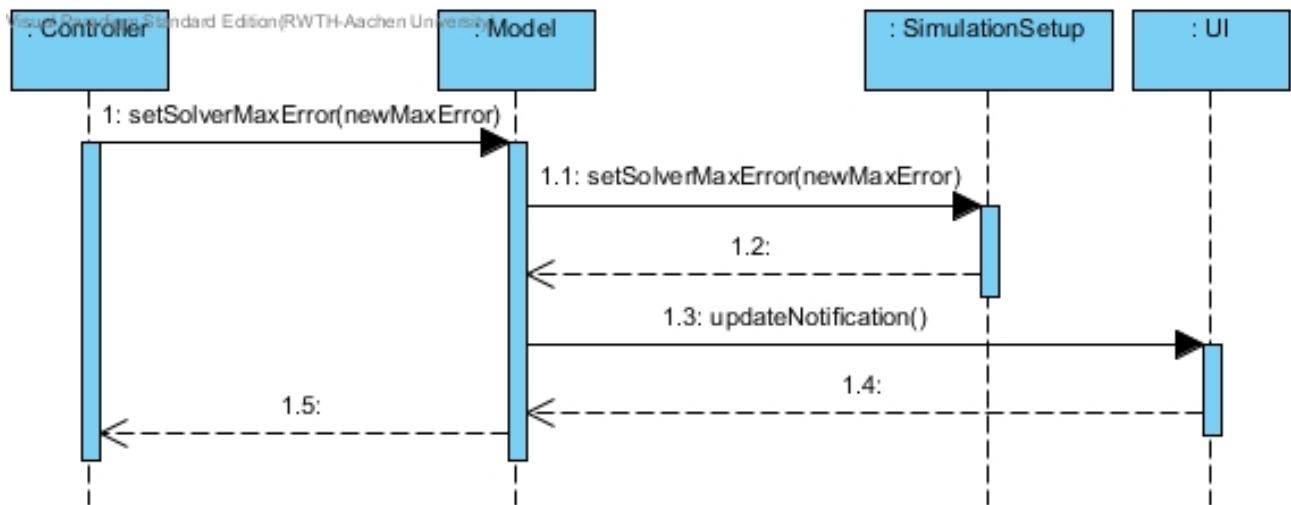


Abbildung 3.31: Sequenzdiagramm Model::setSolverMaxError

### **setSolverMaxIt**

Das Sequenzdiagramm für *setSolverMaxIt* ist in Abbildung 3.32 dargestellt. *setSolverMaxIt* setzt die maximale Iterationsanzahl des LGS Lözers.

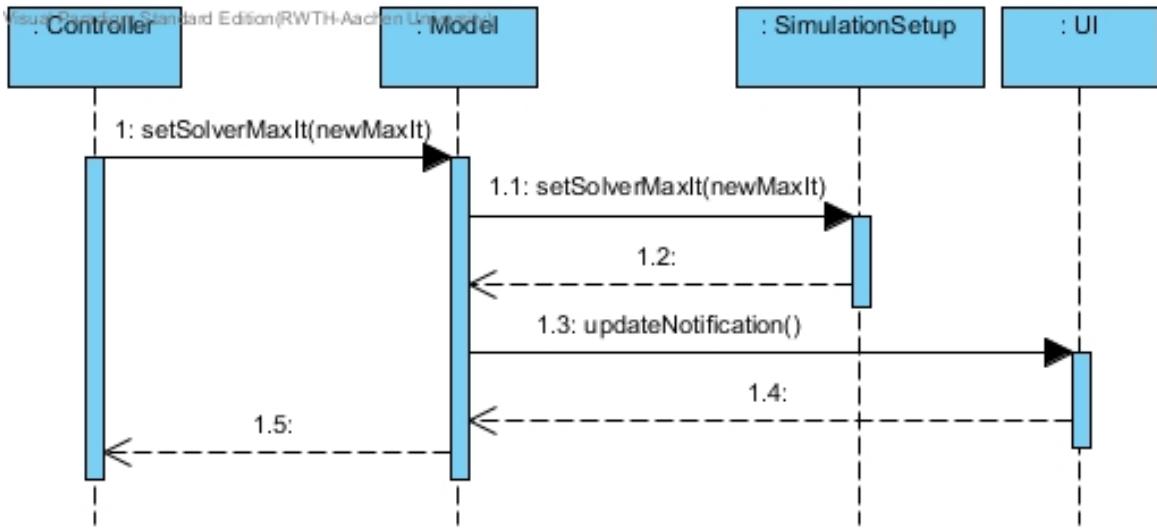


Abbildung 3.32: Sequenzdiagramm Model::setSolverMaxIt

### setT

Das Sequenzdiagramm für *setT* ist in Abbildung 3.33 dargestellt. *setT* setzt den Endzeitpunkt für die Simulation.

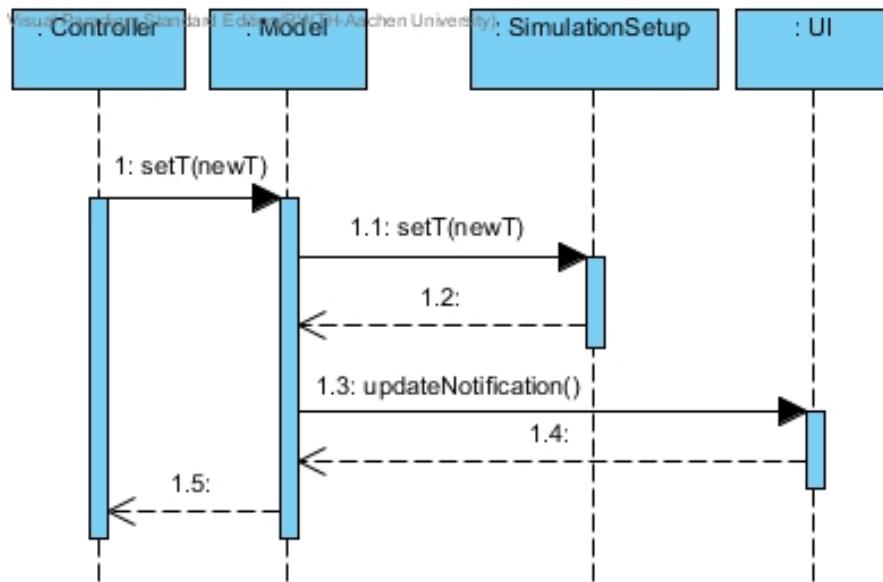


Abbildung 3.33: Sequenzdiagramm Model::setT

### setUseHeatSources

Das Sequenzdiagramm für *setUseHeatSources* ist in Abbildung 3.34 dargestellt. *setUseHeatSources* aktiviert die Nutzung der Wärmequellen-Gebiete für die Optimierung.

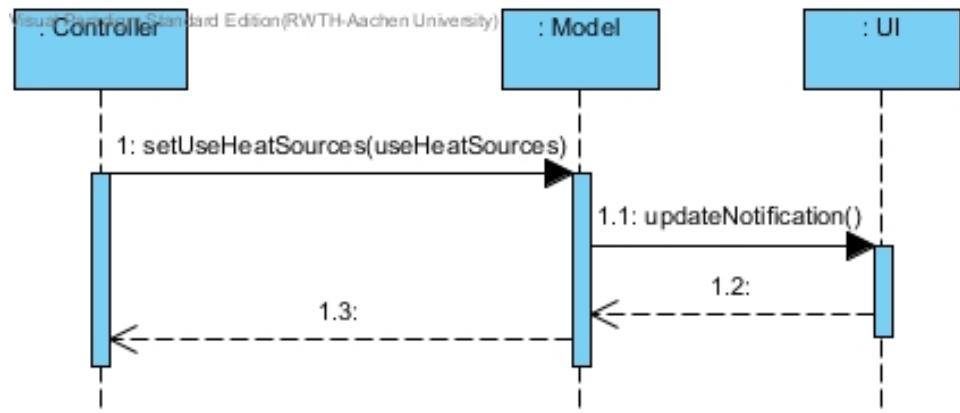


Abbildung 3.34: Sequenzdiagramm Model::setUseHeatSources

### simulate

Das Sequenzdiagramm für `simulate` ist in Abbildung 3.35 dargestellt. `simulate` startet eine Simulation der Wärmeleitungsgleichung.

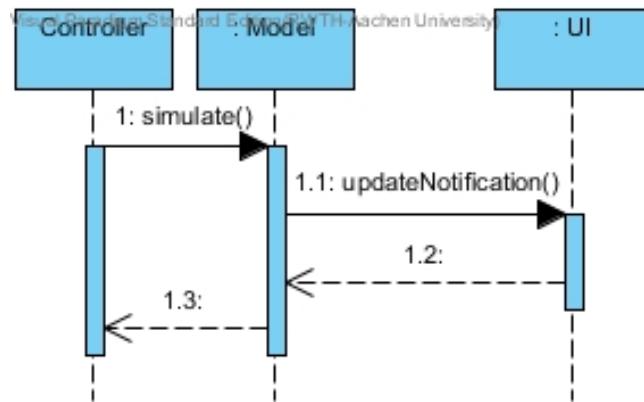


Abbildung 3.35: Sequenzdiagramm Model::simulate

### updateAreaValue

Das Sequenzdiagramm für `updateAreaValue` ist in Abbildung 3.36 dargestellt. `updateAreaValue` ändert den Wert eines Gebietes vom übergebenen Typ.

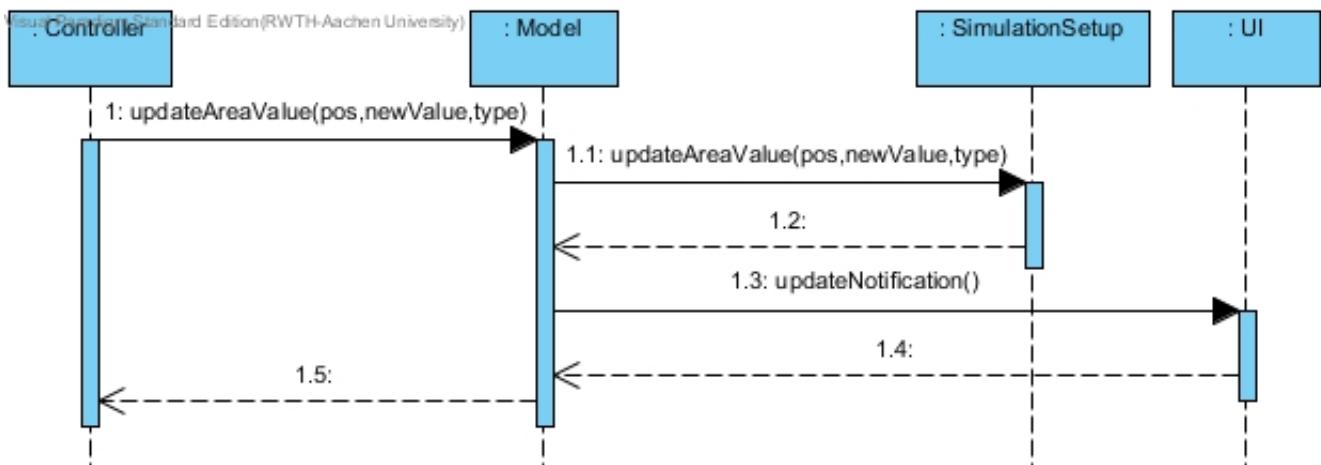


Abbildung 3.36: Sequenzdiagramm Model::updateAreaValue

### 3.2.2.2 SimulationSetup

#### getContainingAreaID

Das Sequenzdiagramm für *getContainingAreaID* ist in Abbildung 3.37 dargestellt. *getContainingAreaID* findet die ID des ersten Gebietes, das den übergebenen Punkt enthält.

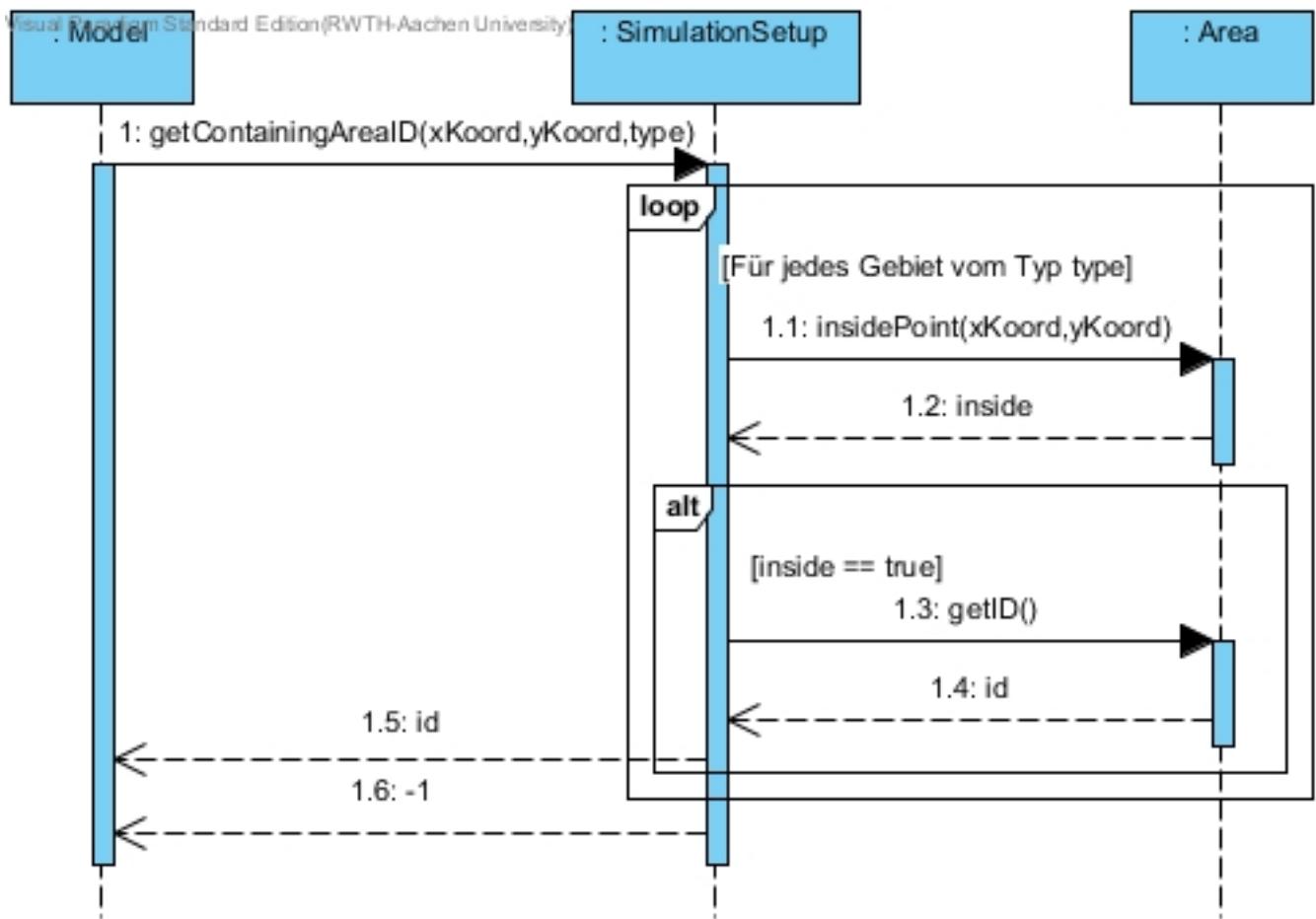


Abbildung 3.37: Sequenzdiagramm SimulationSetup::getContainingAreaID

#### updateAreaValue

Das Sequenzdiagramm für *updateAreaValue* ist in Abbildung 3.38 dargestellt. *updateAreaValue* ändert den Wert des Gebietes vom übergebenen Typ an der übergebenen Position.

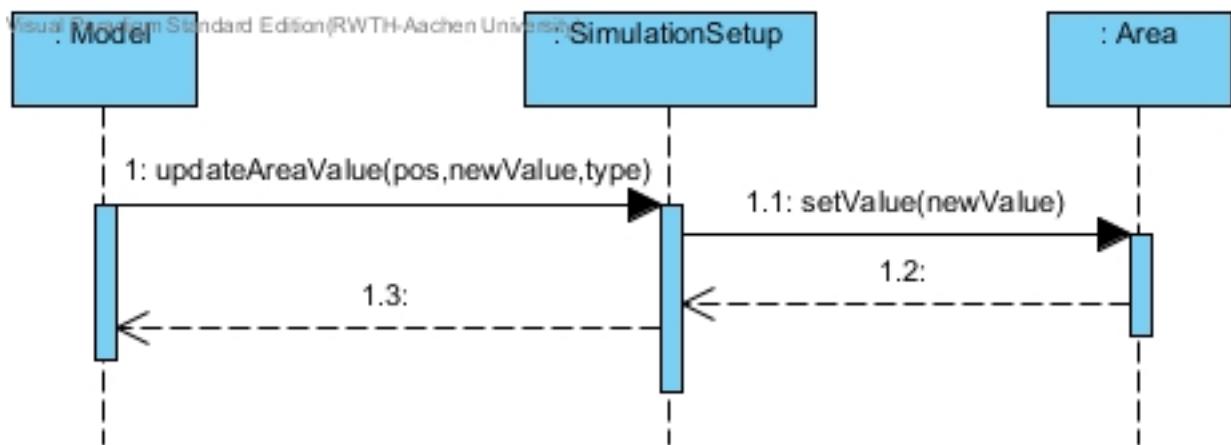


Abbildung 3.38: Sequenzdiagramm SimulationSetup::updateAreaValue

### 3.2.2.3 SimulationWorker

#### simpleSimulation

Das Sequenzdiagramm für *simpleSimulation* ist in Abbildung 3.39 dargestellt. *simpleSimulation* führt eine Simulation der Wärmeleitungsgleichung ohne Speichern der Zwischenschritte (für die Optimierung).

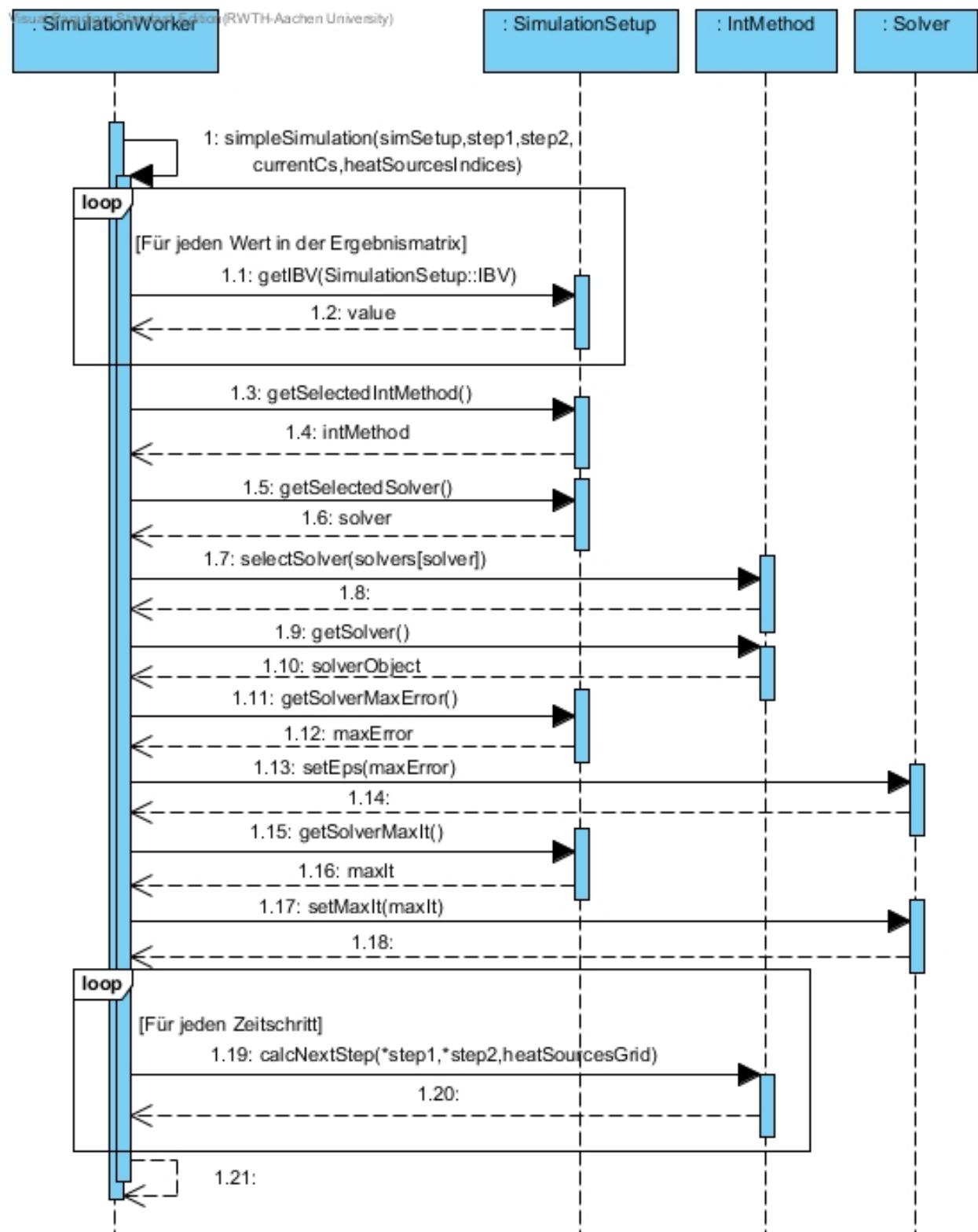


Abbildung 3.39: Sequenzdiagramm SimulationWorker::simpleSimulation

## startOptimizationSlot

Das Sequenzdiagramm für *startOptimizationSlot* ist in Abbildung 3.40 dargestellt. *startOptimizationSlot* führt eine Optimierung der Temperaturleitkoeffizienten durch.

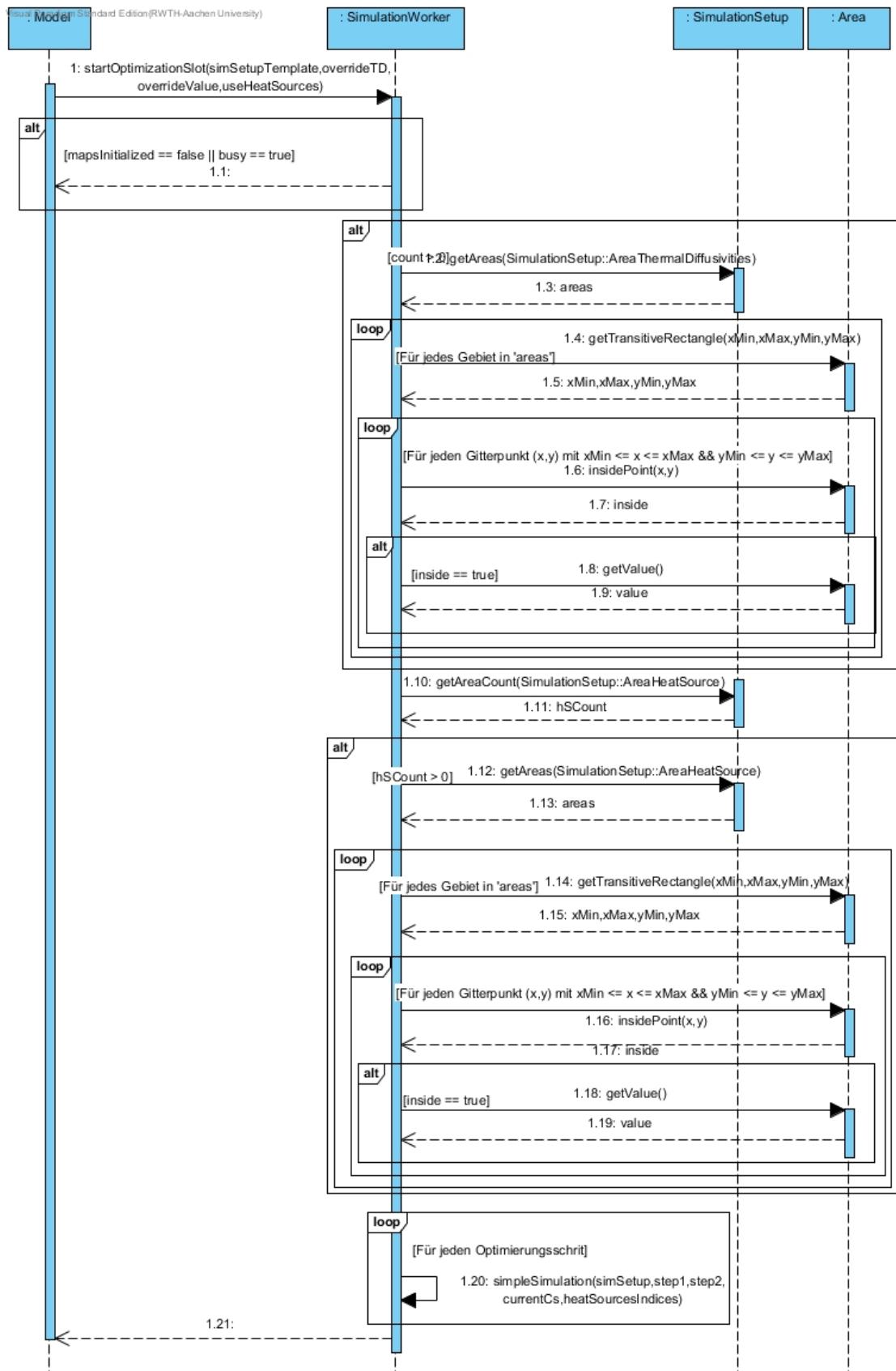


Abbildung 3.40: Sequenzdiagramm *SimulationWorker::startOptimizationSlot*

## startSimulationSlot

Das Sequenzdiagramm für *startSimulationSlot* ist in Abbildung 3.41 dargestellt. *startSimulationSlot* führt eine Simulation der Wärmeleitungsgleichung durch.

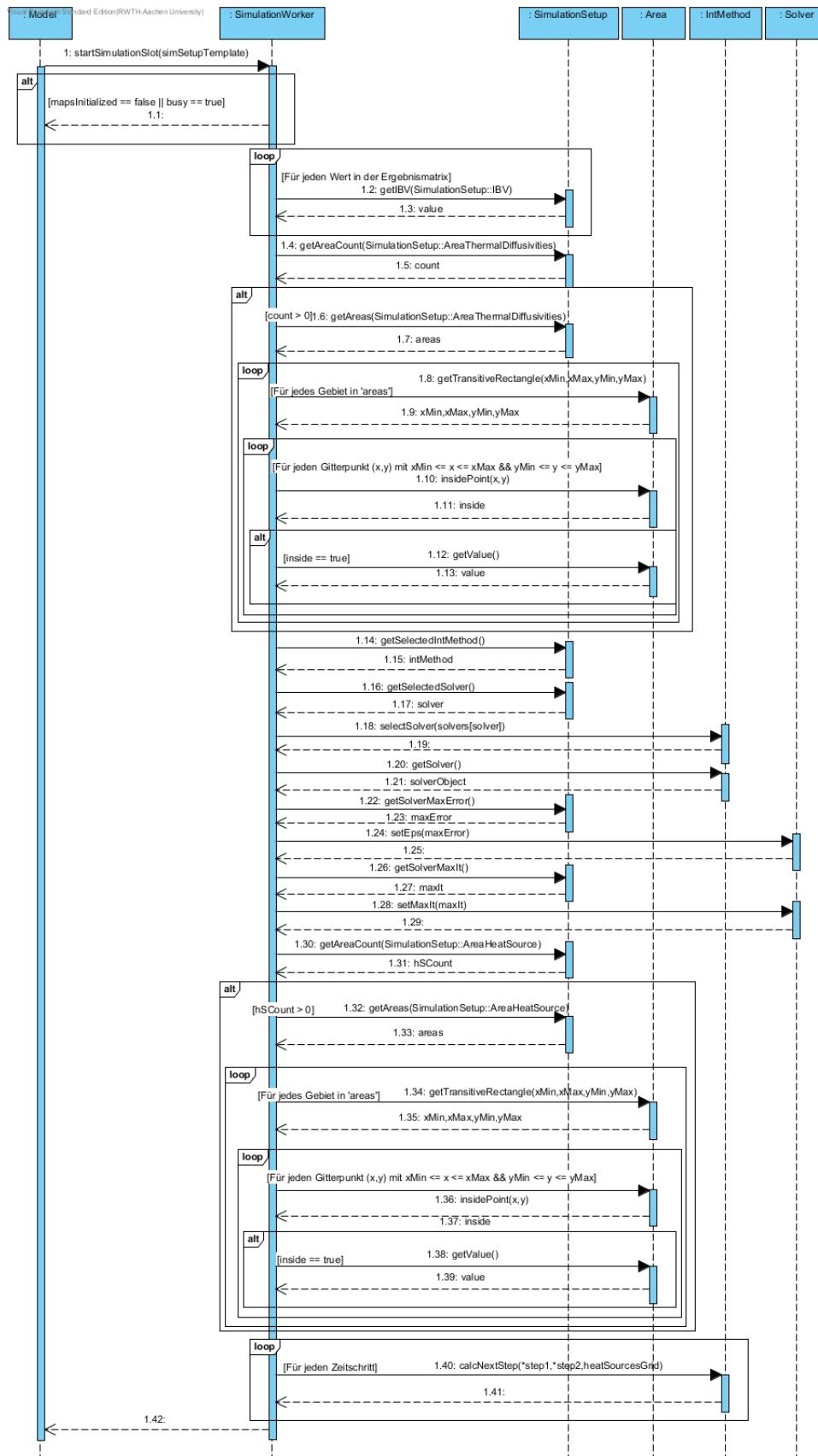


Abbildung 3.41: Sequenzdiagramm SimulationWorker::startSimulationSlot

### 3.2.3 Paket presentation

Das Klassendiagramm in Abbildung 3.42 zeigt alle im Paket *presentation* enthaltene Klassen.

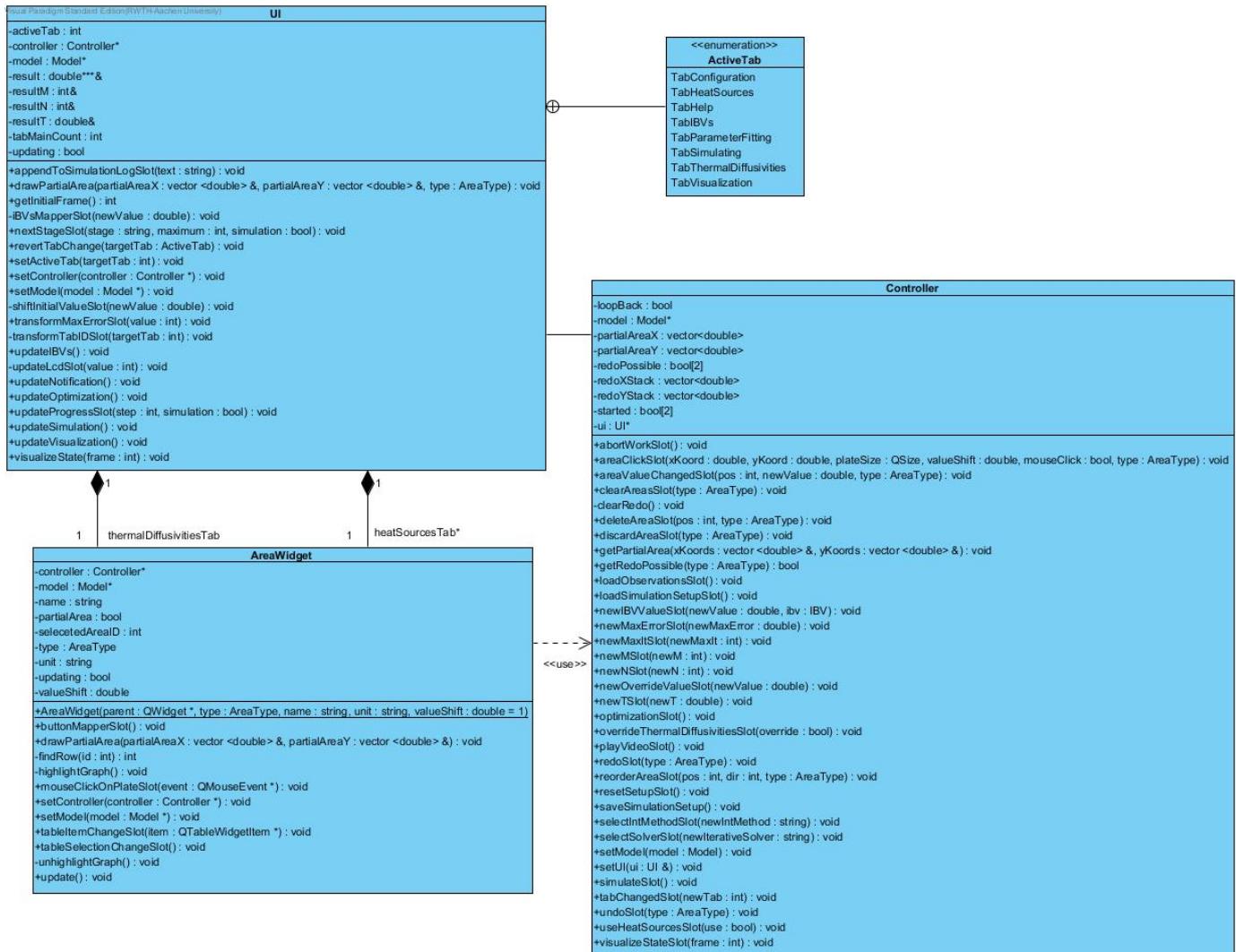


Abbildung 3.42: Klassendiagramm presentation

#### 3.2.3.1 AreaWidget

##### drawPartialArea

Das Sequenzdiagramm für *drawPartialArea* ist in Abbildung 3.43 dargestellt. *drawPartialArea* zeichnet ein neues angefangenes Gebiet.

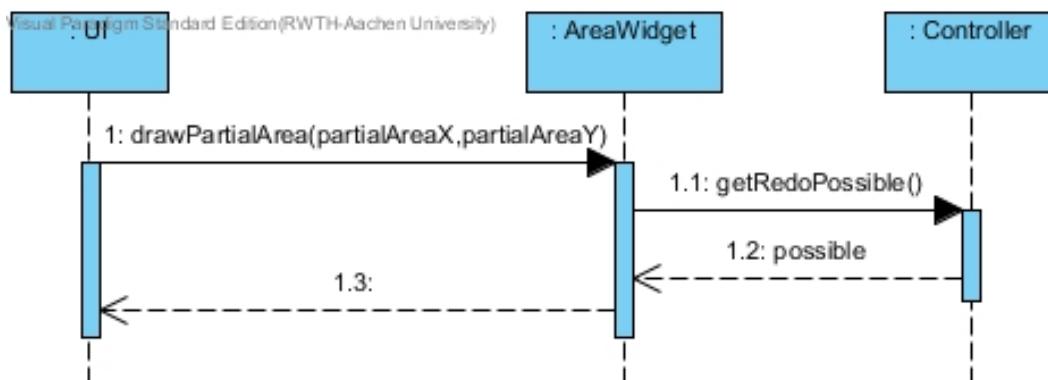


Abbildung 3.43: Sequenzdiagramm AreaWidget::drawPartialArea

## update

Das Sequenzdiagramm für *update* ist in Abbildung 3.44 dargestellt. *update* aktualisiert den Tab zur Gebietseingabe.

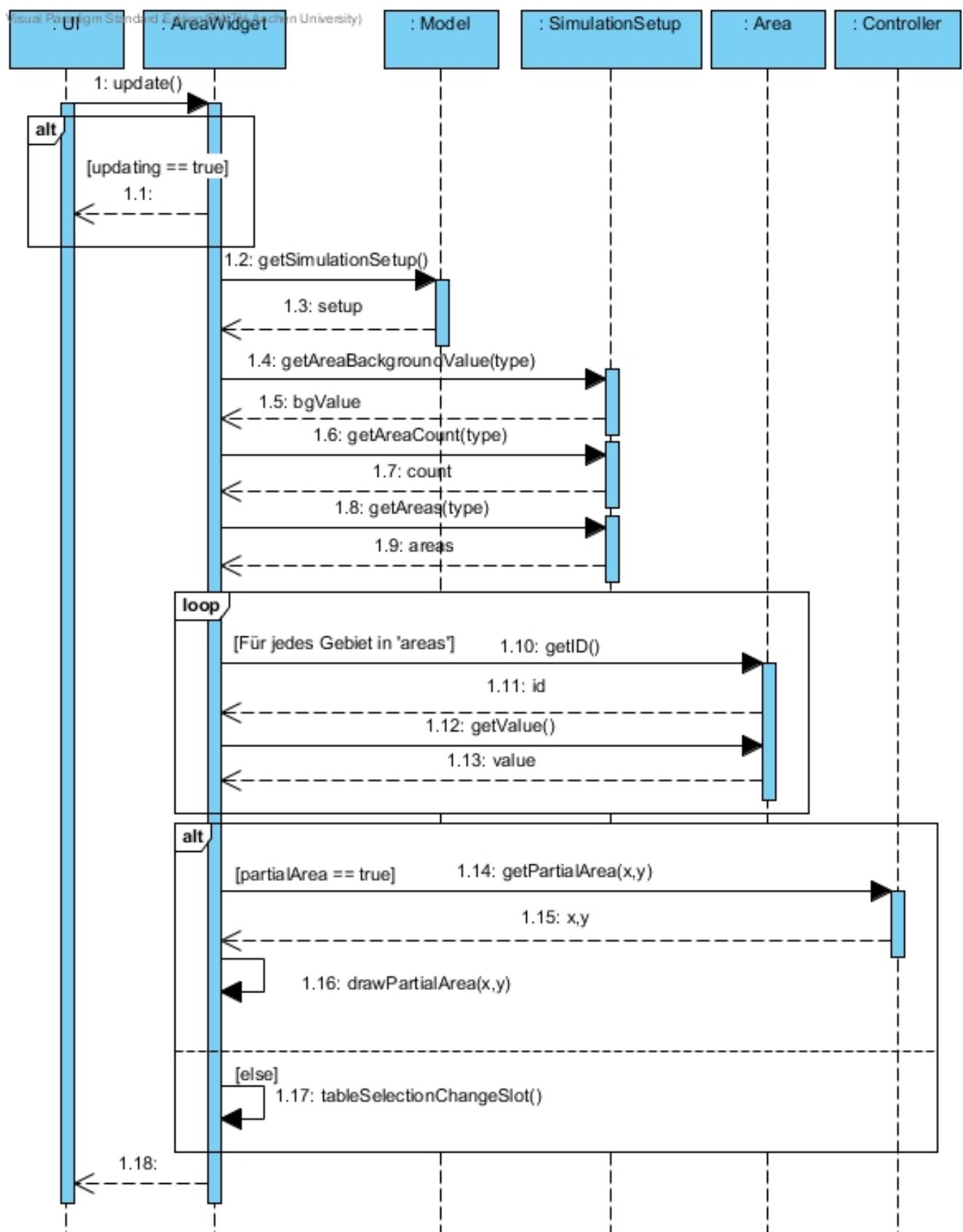


Abbildung 3.44: Sequenzdiagramm `AreaWidget::update`

### 3.2.3.2 Controller

#### abortWorkSlot

Das Sequenzdiagramm für *abortWorkSlot* ist in Abbildung 3.45 dargestellt. *abortWorkSlot* bricht eine laufende Simulation/Optimierung ab.

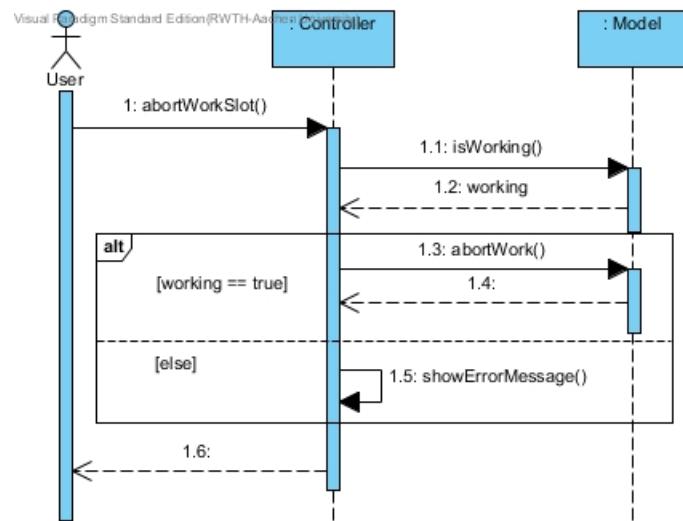


Abbildung 3.45: Sequenzdiagramm Controller::abortWorkSlot

#### areaClickSlot

Das Sequenzdiagramm für *areaClickSlot* ist in Abbildung 3.46 dargestellt. *areaClickSlot* verwaltet Mausklicke auf eine der beiden Gebiet-Platten.

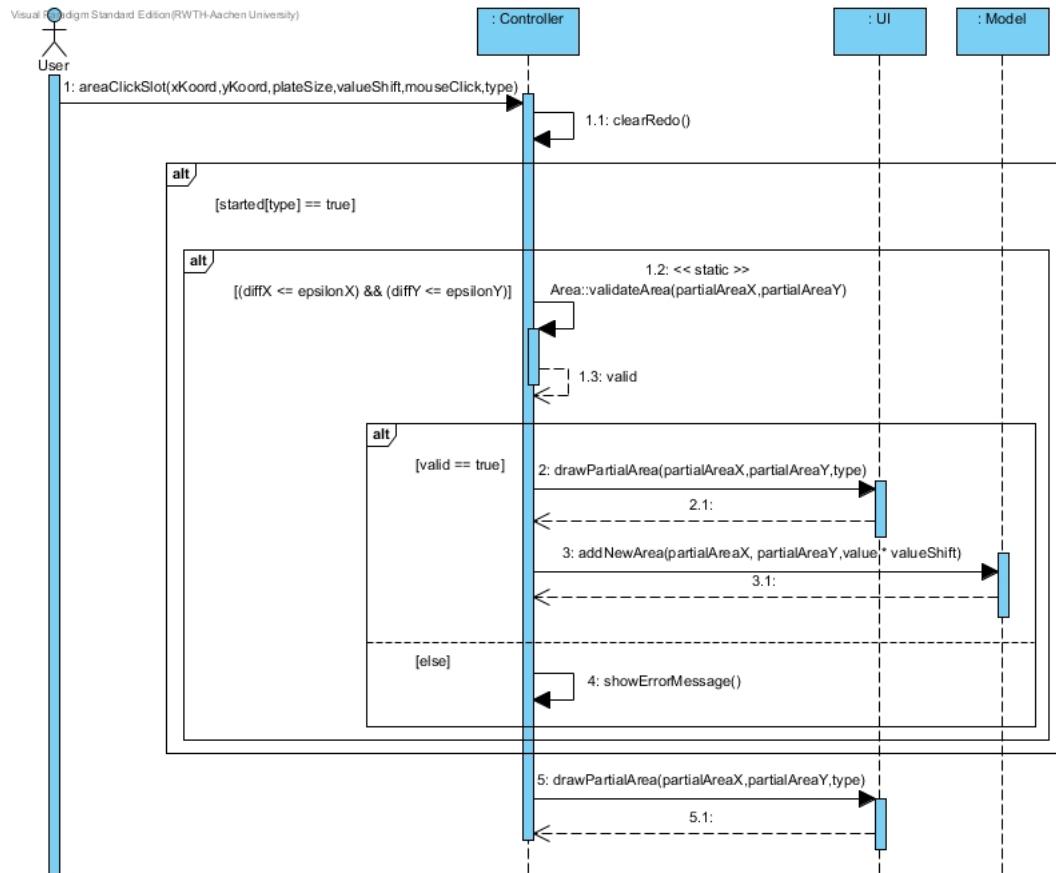


Abbildung 3.46: Sequenzdiagramm Controller::areaClickSlot

## areaValueChangedSlot

Das Sequenzdiagramm für *areaValueChangedSlot* ist in Abbildung 3.47 dargestellt. *areaValueChangedSlot* verwaltet das Ändern der Werte der verschiedenen Gebiete.

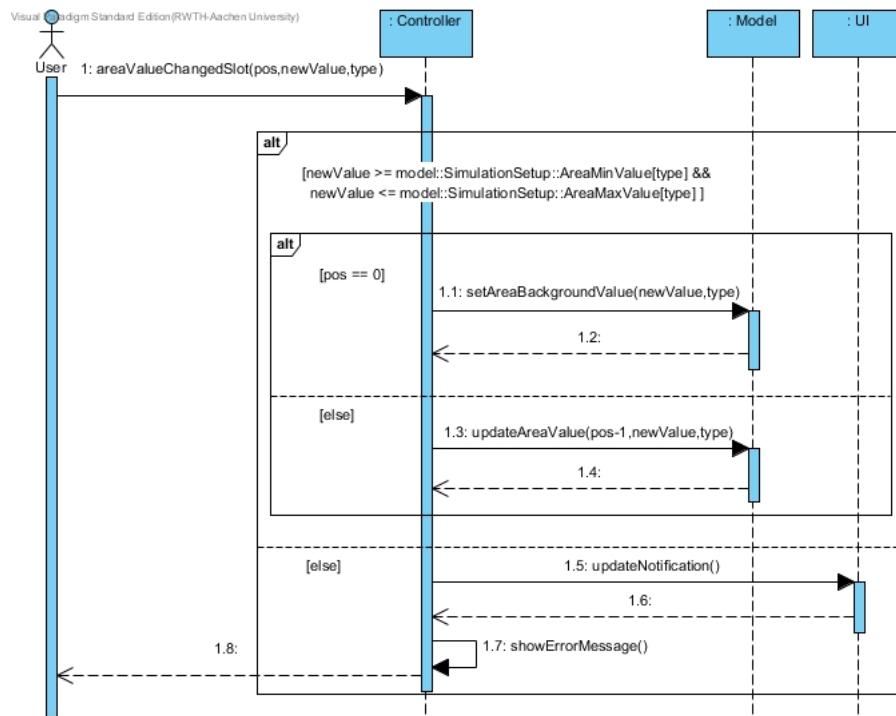


Abbildung 3.47: Sequenzdiagramm Controller::areaValueChangedSlot

## clearAreasSlot

Das Sequenzdiagramm für *clearAreasSlot* ist in Abbildung 3.48 dargestellt. *clearAreasSlot* löscht alle Gebiete eines Typs.

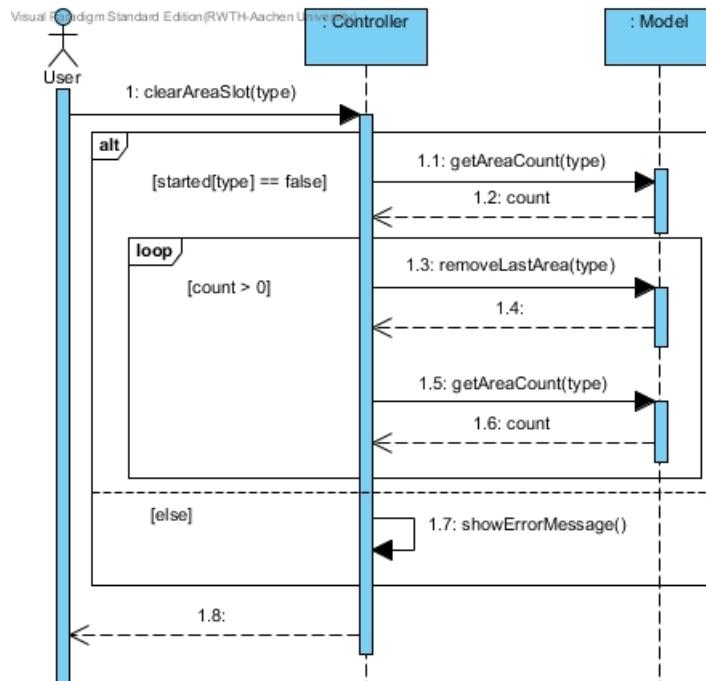


Abbildung 3.48: Sequenzdiagramm Controller::clearAreasSlot

## deleteAreaSlot

Das Sequenzdiagramm für *deleteAreaSlot* ist in Abbildung 3.49 dargestellt. *deleteAreaSlot* löscht ein Gebiet eines Types.

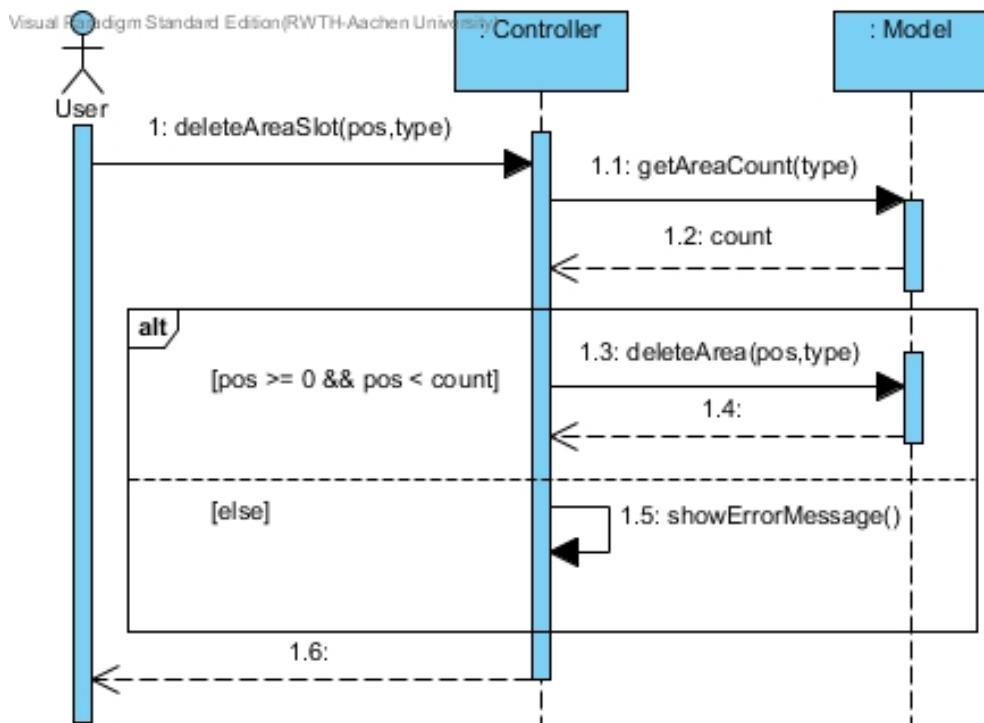


Abbildung 3.49: Sequenzdiagramm Controller::deleteAreaSlot

## discardAreaSlot

Das Sequenzdiagramm für *discardAreaSlot* ist in Abbildung 3.50 dargestellt. *discardAreaSlot* bricht das Erstellen eines neuen Gebietes ab.

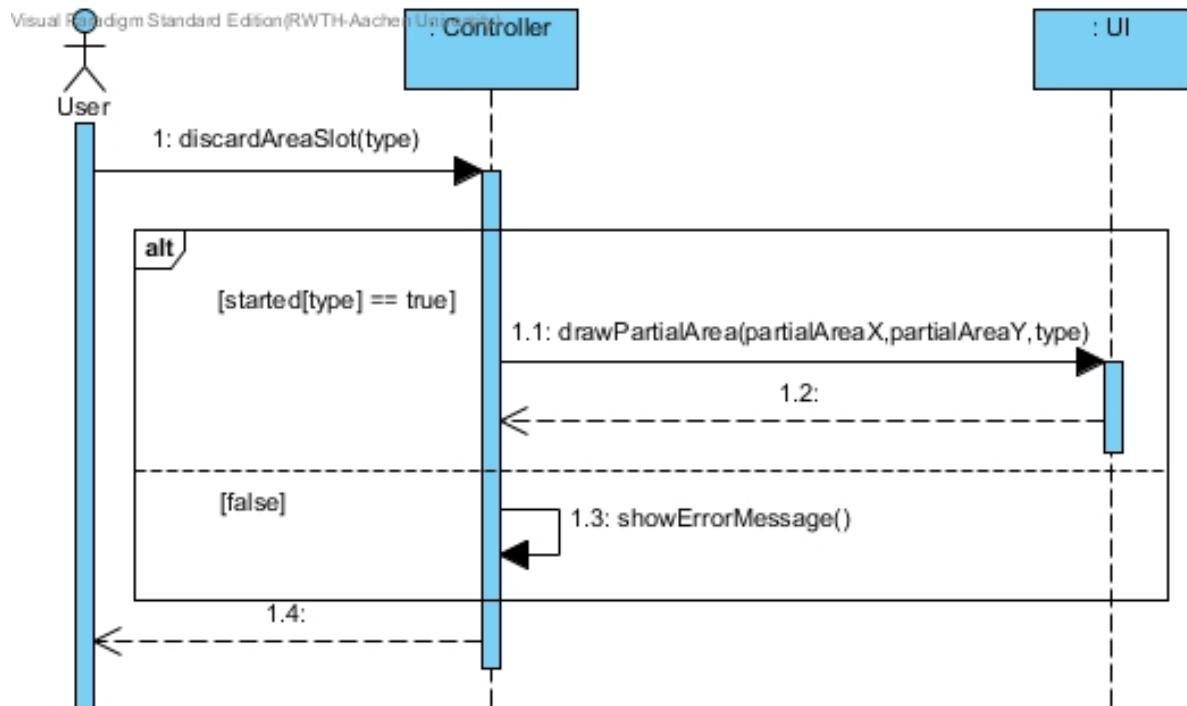


Abbildung 3.50: Sequenzdiagramm Controller::discardAreaSlot

## loadObservationsSlot

Das Sequenzdiagramm für *loadObservationsSlot* ist in Abbildung 3.51 dargestellt. *loadObservationsSlot* verwaltet das Einlesen von Messdaten für die Optimierung.

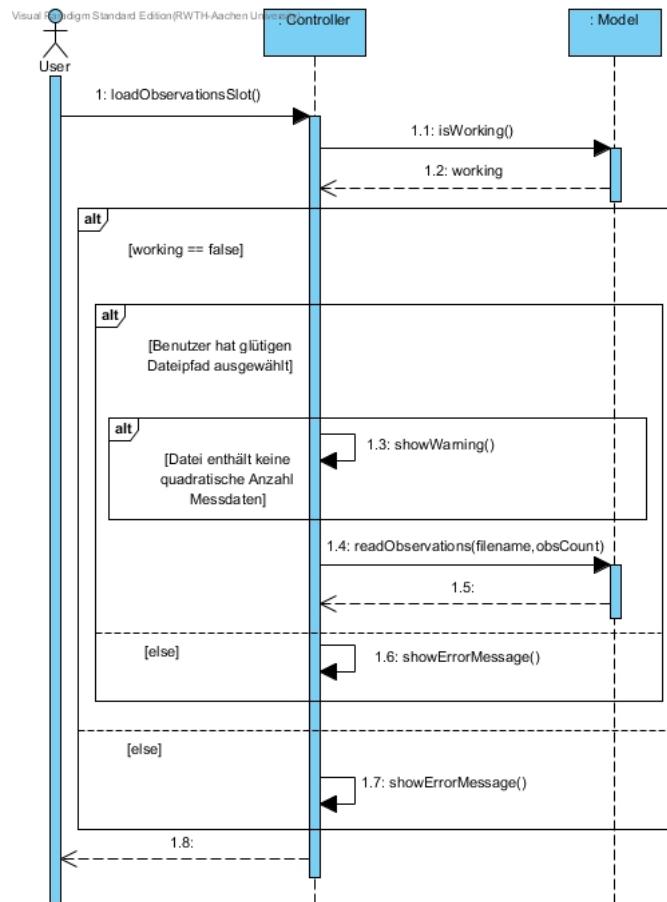


Abbildung 3.51: Sequenzdiagramm Controller::loadObservationsSlot

## loadSimulationSetupSlot

Das Sequenzdiagramm für *loadSimulationSetupSlot* ist in Abbildung 3.52 dargestellt. *loadSimulationSetupSlot* verwaltet das Einlesen von gespeicherten Simulationseinstellungen.

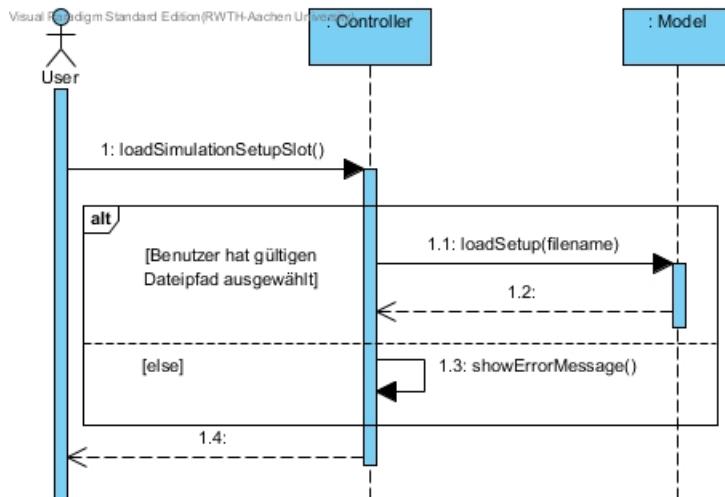


Abbildung 3.52: Sequenzdiagramm Controller::loadSimulationSetupSlot

## newIBVValueSlot

Das Sequenzdiagramm für *newIBVValueSlot* ist in Abbildung 3.53 dargestellt. *newIBVValueSlot* verwaltet das Ändern eines Rand- bzw. des Anfangswertes.

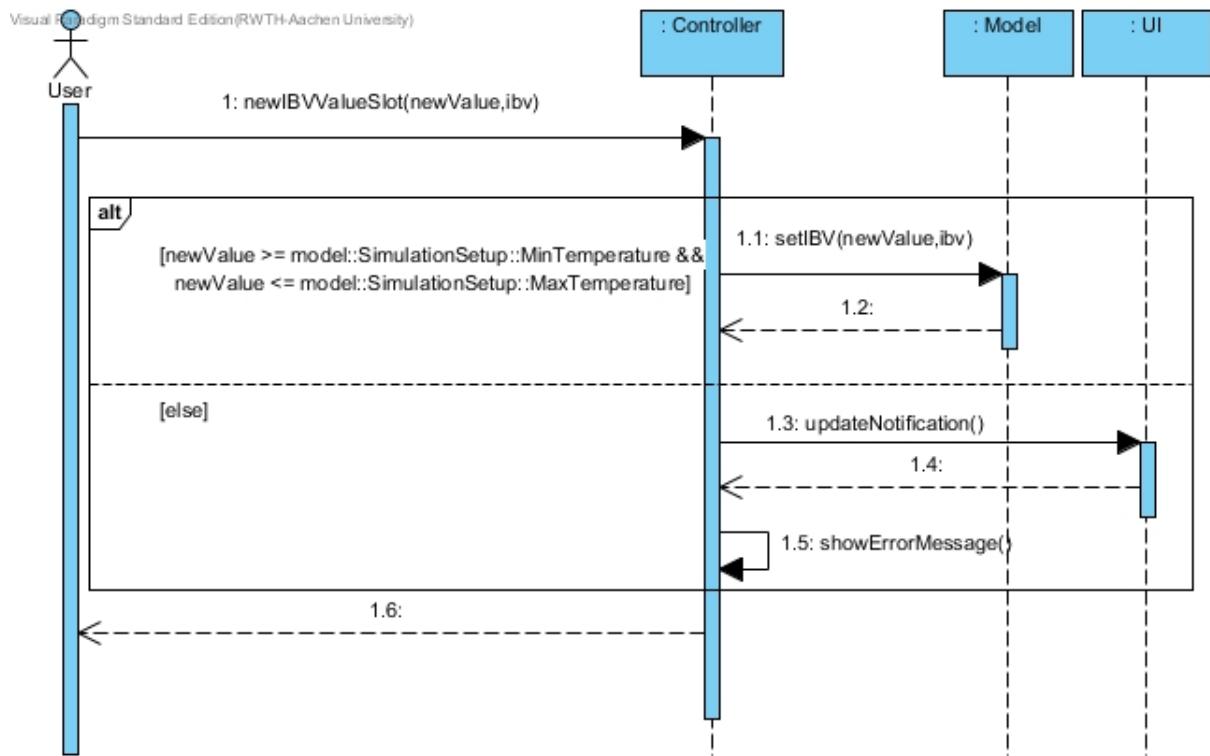


Abbildung 3.53: Sequenzdiagramm Controller::newIBVValueSlot

## newMaxErrorSlot

Das Sequenzdiagramm für *newMaxErrorSlot* ist in Abbildung 3.54 dargestellt. *newMaxErrorSlot* verwaltet das Ändern der Fehlertoleranz des LGS Lösen.

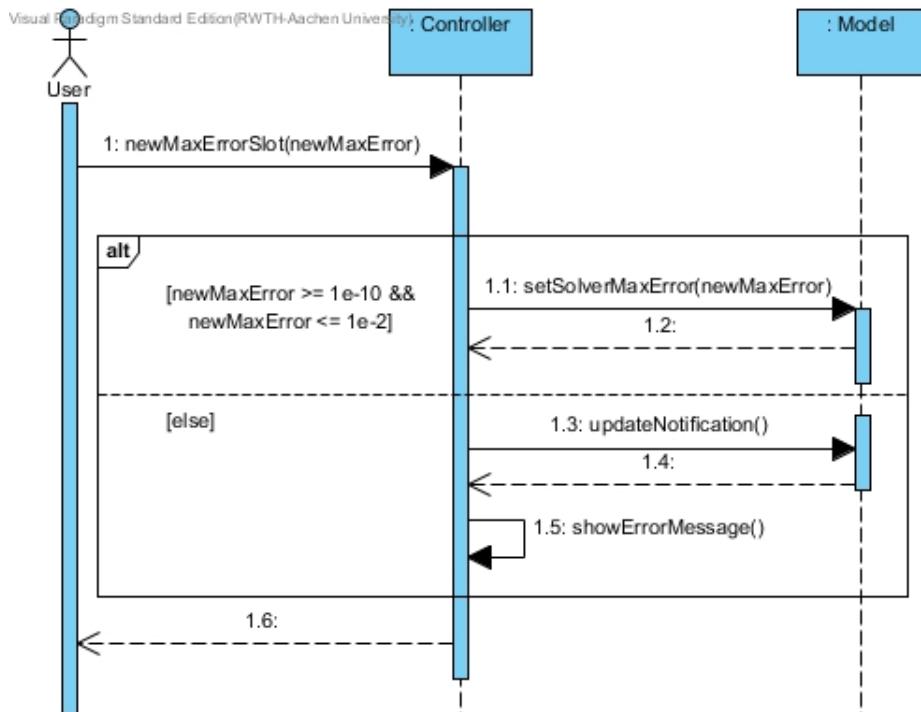


Abbildung 3.54: Sequenzdiagramm Controller::newMaxErrorSlot

## newMaxItSlot

Das Sequenzdiagramm für *newMaxItSlot* ist in Abbildung 3.55 dargestellt. *newMaxItSlot* verwaltet das Ändern der maximalen Iterationsanzahl des LGS Lösers.

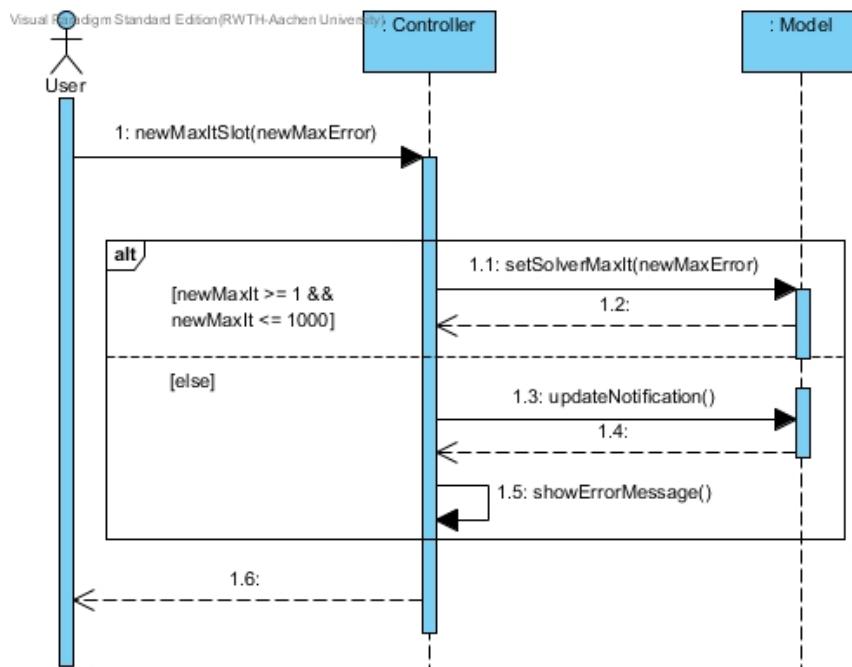


Abbildung 3.55: Sequenzdiagramm Controller::newMaxItSlot

## newMSlot

Das Sequenzdiagramm für *newMSlot* ist in Abbildung 3.56 dargestellt. *newMSlot* verwaltet das Ändern der Zeitdiskretisierungsgröße für die Simulation.

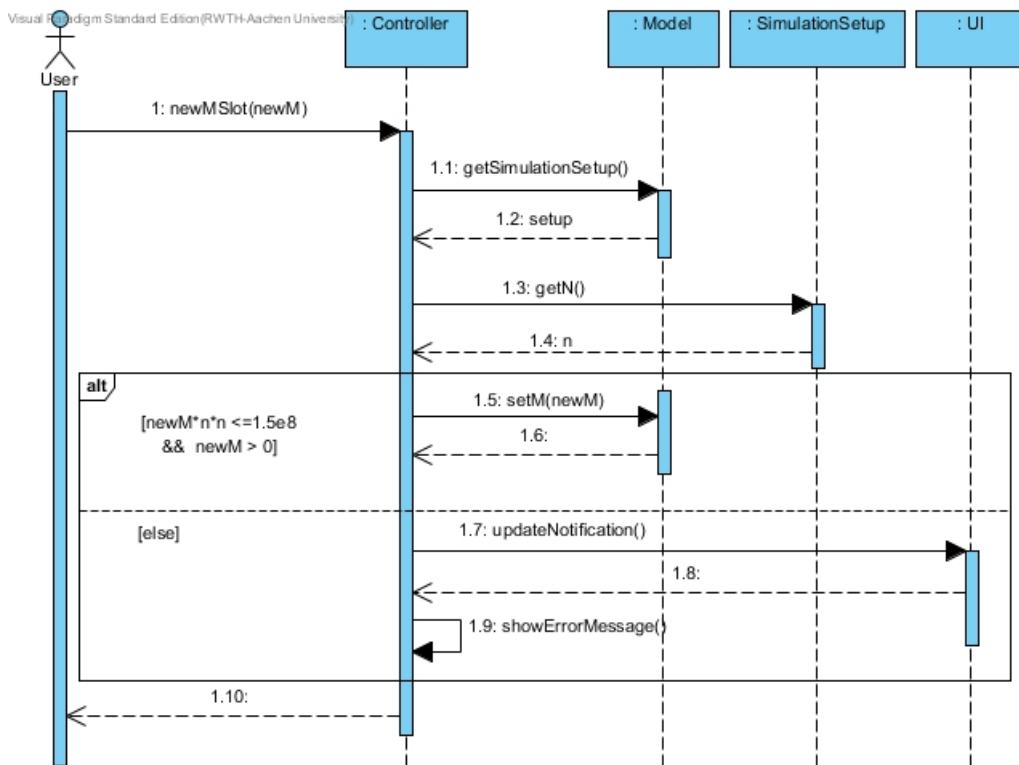


Abbildung 3.56: Sequenzdiagramm Controller::newMSlot

## newNSlot

Das Sequenzdiagramm für *newNSlot* ist in Abbildung 3.57 dargestellt. *newNSlot* verwaltet das Ändern der Ortsdiskretisierungsgröße für die Simulation.

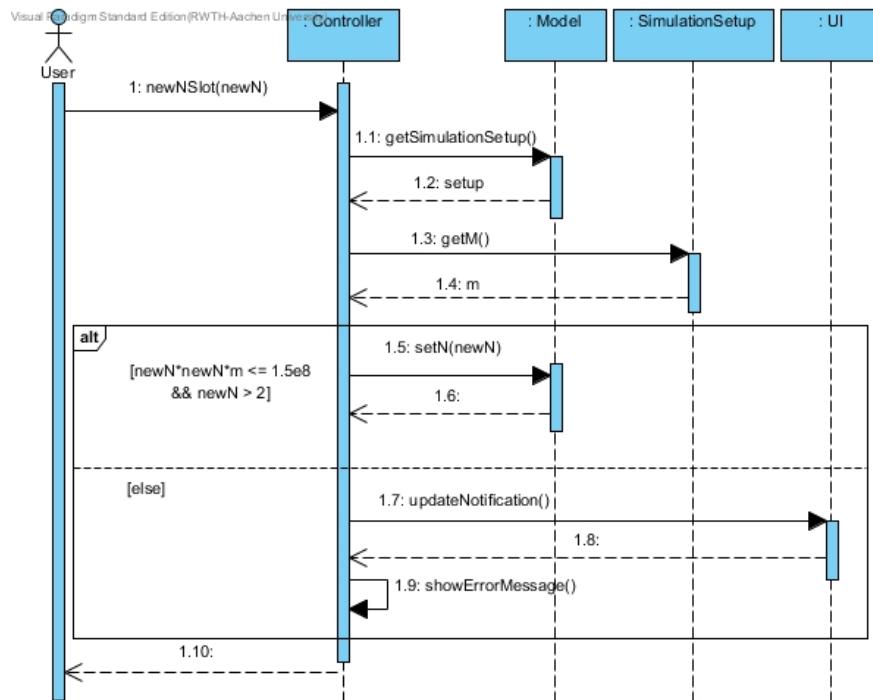


Abbildung 3.57: Sequenzdiagramm Controller::newNSlot

## newOverrideValueSlot

Das Sequenzdiagramm für *newOverrideValueSlot* ist in Abbildung 3.58 dargestellt. *newOverrideValueSlot* verwaltet das Ändern des manuellen Anfangswertes für die Optimierung.

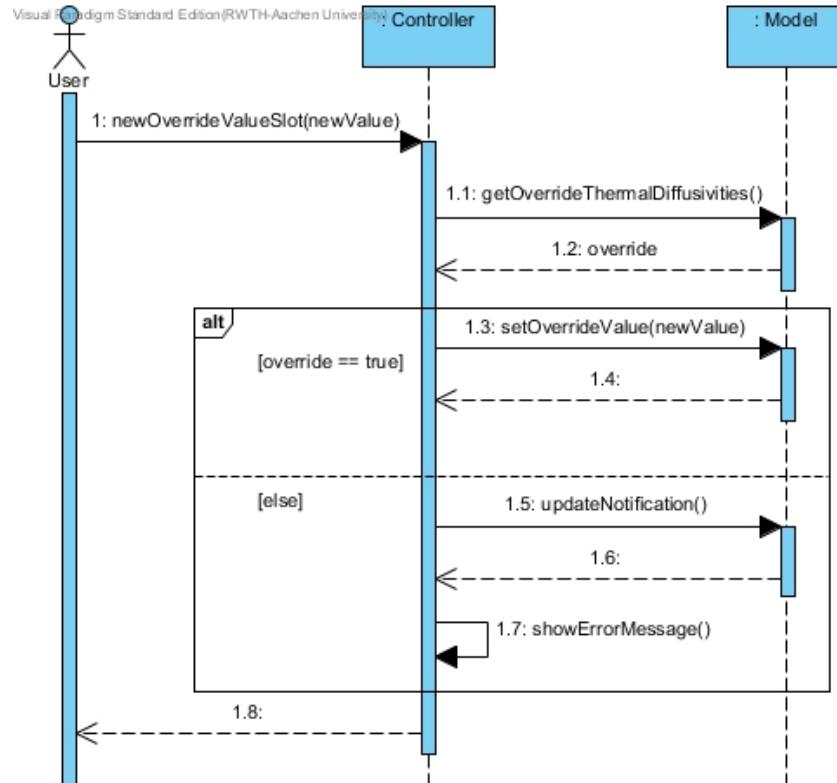


Abbildung 3.58: Sequenzdiagramm Controller::newOverrideValueSlot

## newTSlot

Das Sequenzdiagramm für *newTSlot* ist in Abbildung 3.59 dargestellt. *newTSlot* verwaltet das Ändern des Endzeitpunktes für die Simulation.

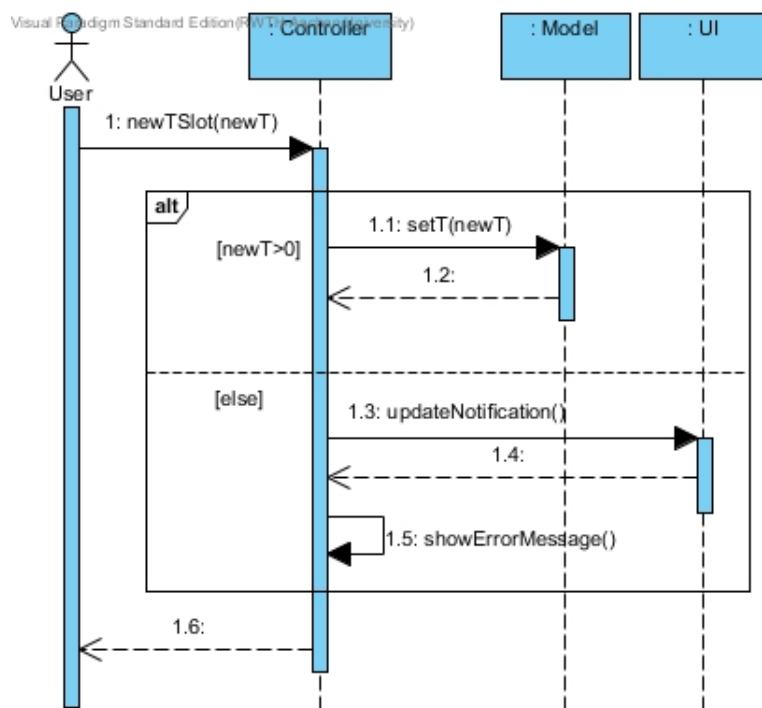


Abbildung 3.59: Sequenzdiagramm Controller::newTSlot

## optimizationSlot

Das Sequenzdiagramm für *optimizationSlot* ist in Abbildung 3.60 dargestellt. *optimizationSlot* verwaltet das Durchführen einer Optimierung der Temperaturleitkoeffizienten.

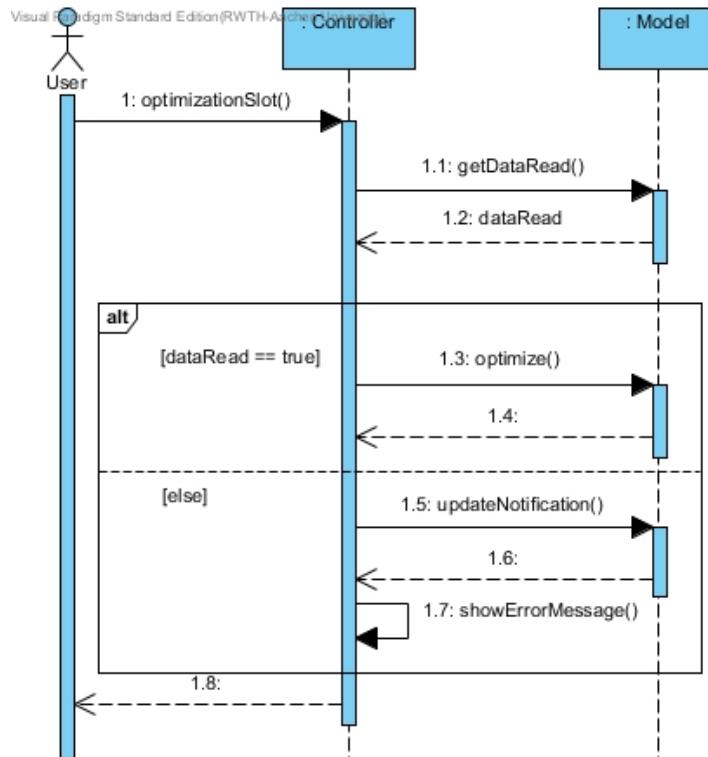


Abbildung 3.60: Sequenzdiagramm Controller::optimizationSlot

## overrideThermalDiffusivitiesSlot

Das Sequenzdiagramm für *overrideThermalDiffusivitiesSlot* ist in Abbildung 3.61 dargestellt. *overrideThermalDiffusivitiesSlot* verwaltet das Aktivieren des manuellen Anfangswertes für die Optimierung.

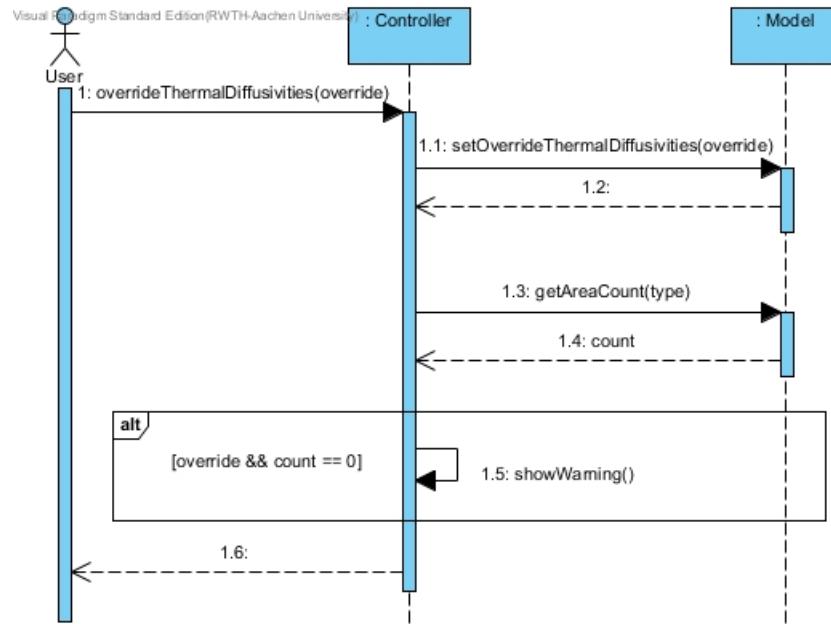


Abbildung 3.61: Sequenzdiagramm Controller::overrideThermalDiffusivitiesSlot

## playVideoSlot

Das Sequenzdiagramm für *playVideoSlot* ist in Abbildung 3.62 dargestellt. *playVideoSlot* visualisiert das Ergebnis der letzten Simulation in Form eines Videos.

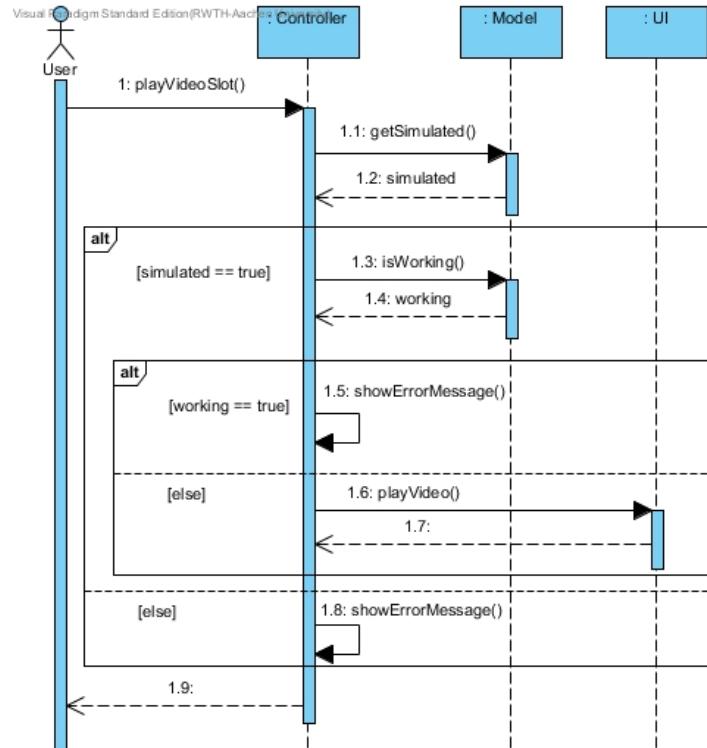


Abbildung 3.62: Sequenzdiagramm Controller::playVideoSlot

## redoSlot

Das Sequenzdiagramm für *redoSlot* ist in Abbildung 3.63 dargestellt. *redoSlot* stellt den letzten rückgängig gemachten Mausklick auf eine der Gebiet-Platten wieder her.

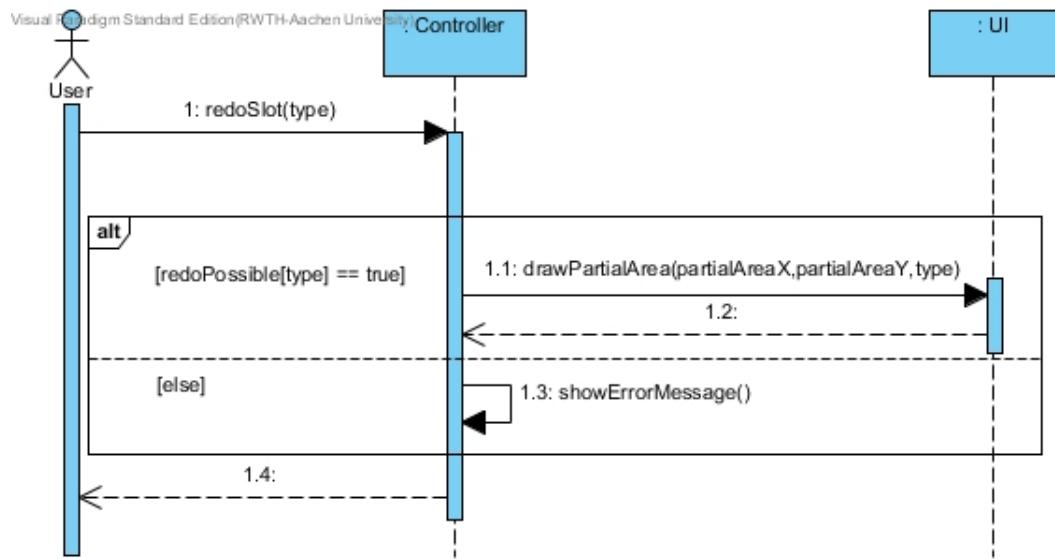


Abbildung 3.63: Sequenzdiagramm Controller::redoSlot

## reorderAreaSlot

Das Sequenzdiagramm für *reorderAreaSlot* ist in Abbildung 3.64 dargestellt. *reorderAreaSlot* verwaltet das Ändern der Reihenfolge der Gebiete.

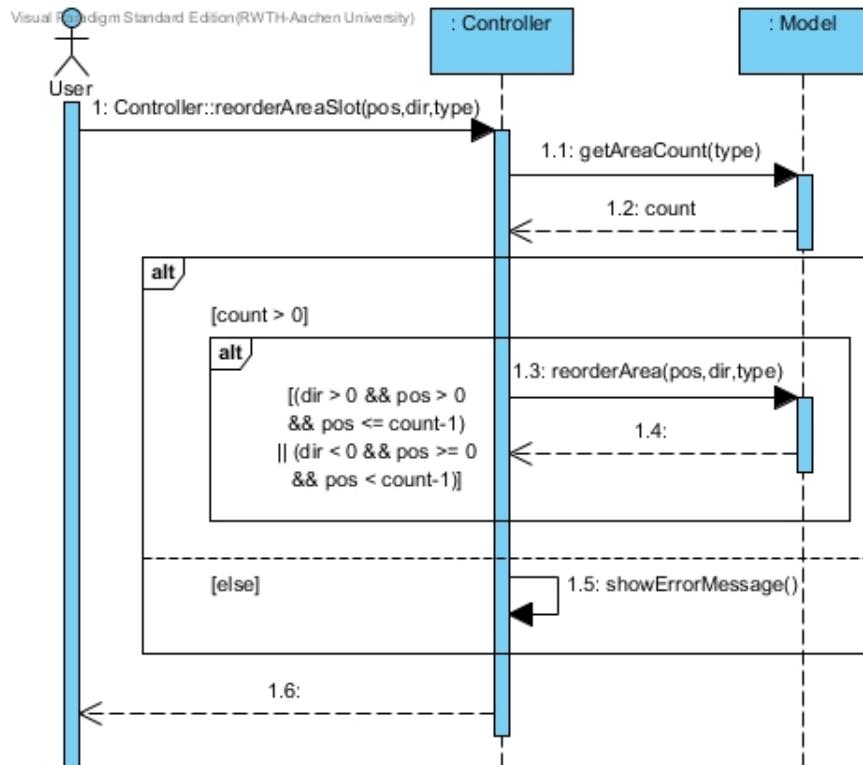


Abbildung 3.64: Sequenzdiagramm Controller::reorderAreaSlot

## resetSimulationSetupSlot

Das Sequenzdiagramm für *resetSimulationSetupSlot* ist in Abbildung 3.65 dargestellt. *resetSimulationSetupSlot* verwaltet das Zurücksetzen der Simulationseinstellungen.

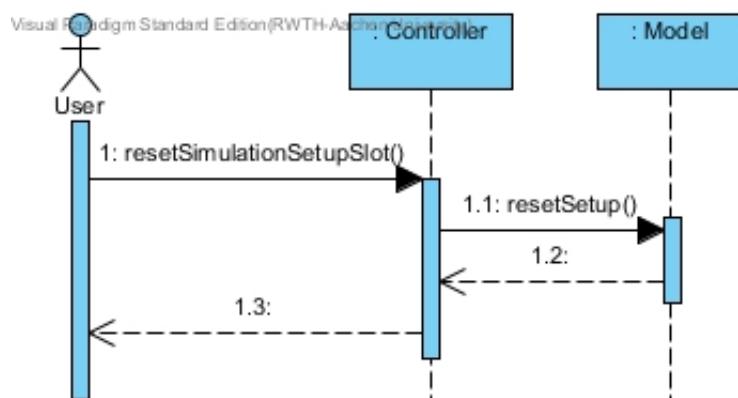


Abbildung 3.65: Sequenzdiagramm Controller::resetSimulationSetupSlot

## saveSimulationSetupSlot

Das Sequenzdiagramm für *saveSimulationSetupSlot* ist in Abbildung 3.66 dargestellt. *saveSimulationSetupSlot* verwaltet das Speichern der Simulationseinstellungen.

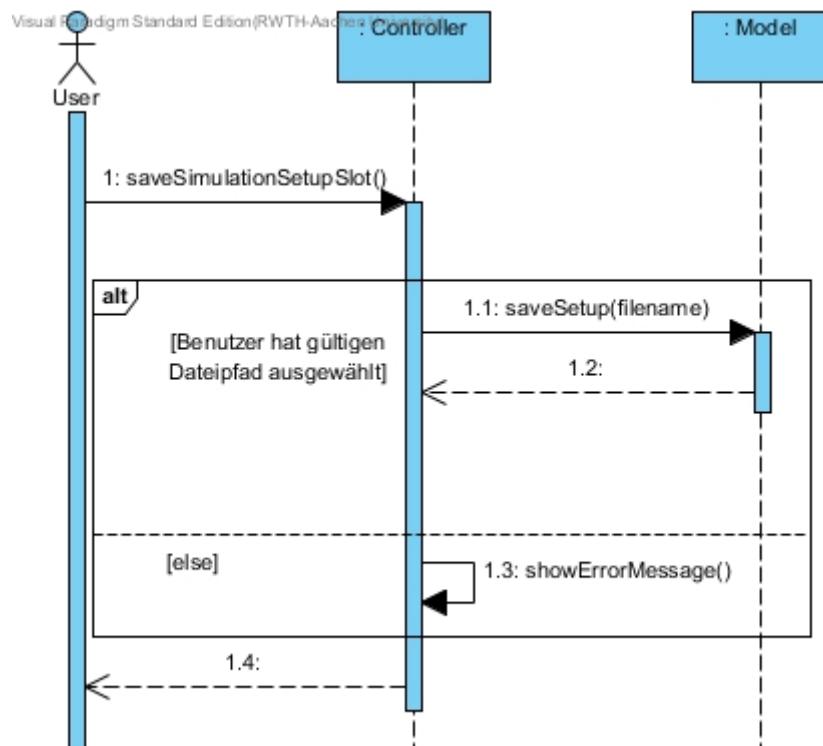


Abbildung 3.66: Sequenzdiagramm Controller::saveSimulationSetupSlot

## selectIntMethodSlot

Das Sequenzdiagramm für *selectIntMethodSlot* ist in Abbildung 3.67 dargestellt. *selectIntMethodSlot* verwaltet das Ändern der Integrationsmethode für die Simulation.

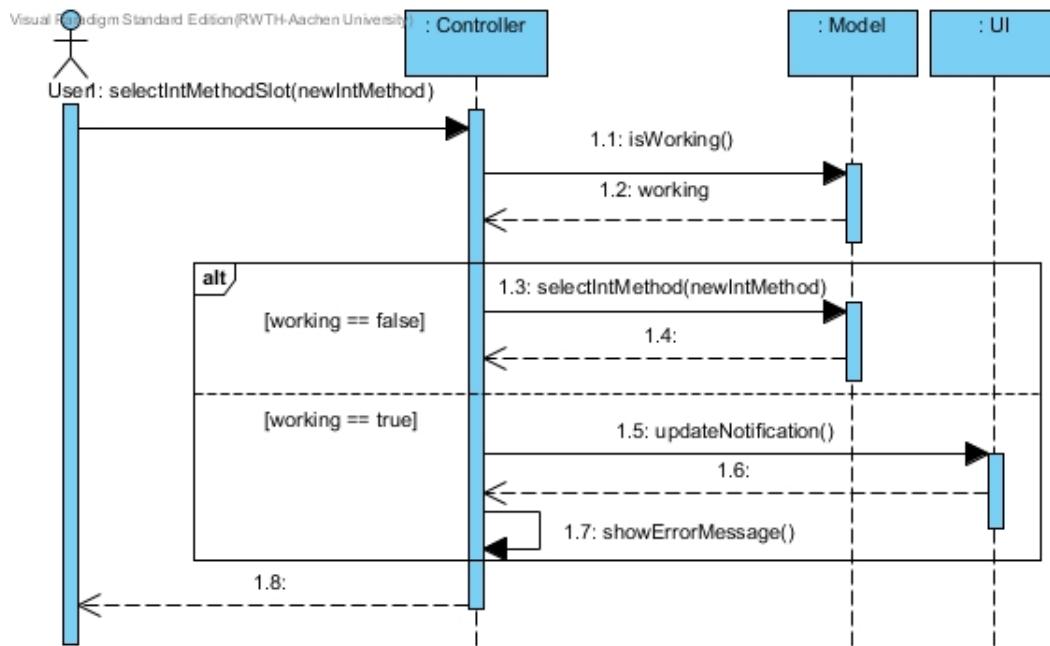


Abbildung 3.67: Sequenzdiagramm Controller::selectIntMethodSlot

## selectSolverSlot

Das Sequenzdiagramm für *selectIterativeSolverSlot* ist in Abbildung 3.68 dargestellt. *selectSolverSlot* verwaltet das Ändern des LGS Lözers für die Simulation.

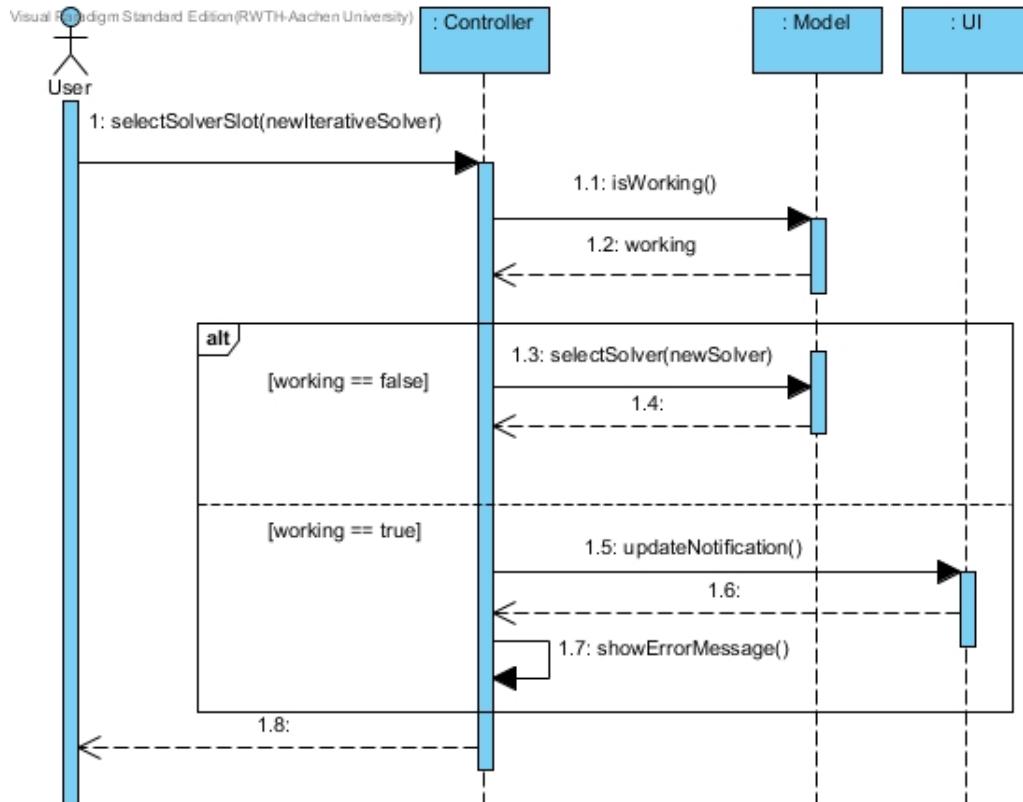


Abbildung 3.68: Sequenzdiagramm Controller::selectIterativeSolverSlot

## simulateSlot

Das Sequenzdiagramm für *simulateSlot* ist in Abbildung 3.69 dargestellt. *selectIntMethodSlot* verwaltet das Durchführen einer Simulation der Wärmeleitungsgleichung.

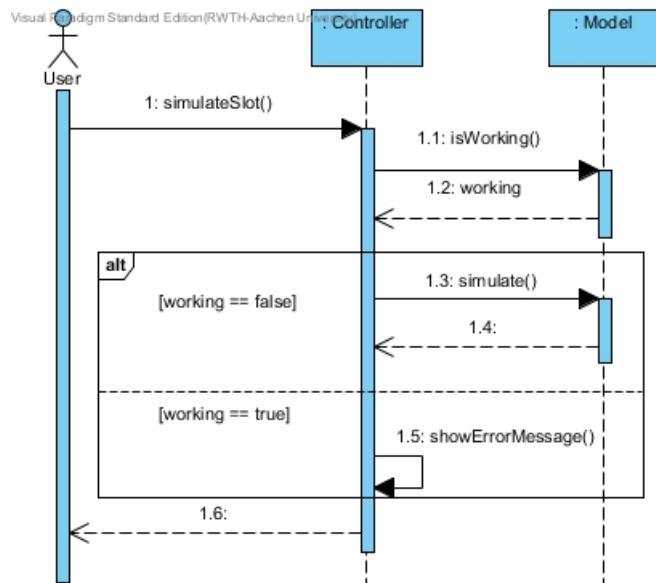


Abbildung 3.69: Sequenzdiagramm Controller::simulateSlot

## tabChangedSlot

Das Sequenzdiagramm für *tabChangedSlot* ist in Abbildung 3.70 dargestellt. *tabChangedSlot* verwaltet das Wechseln des sichtbaren Tabs.

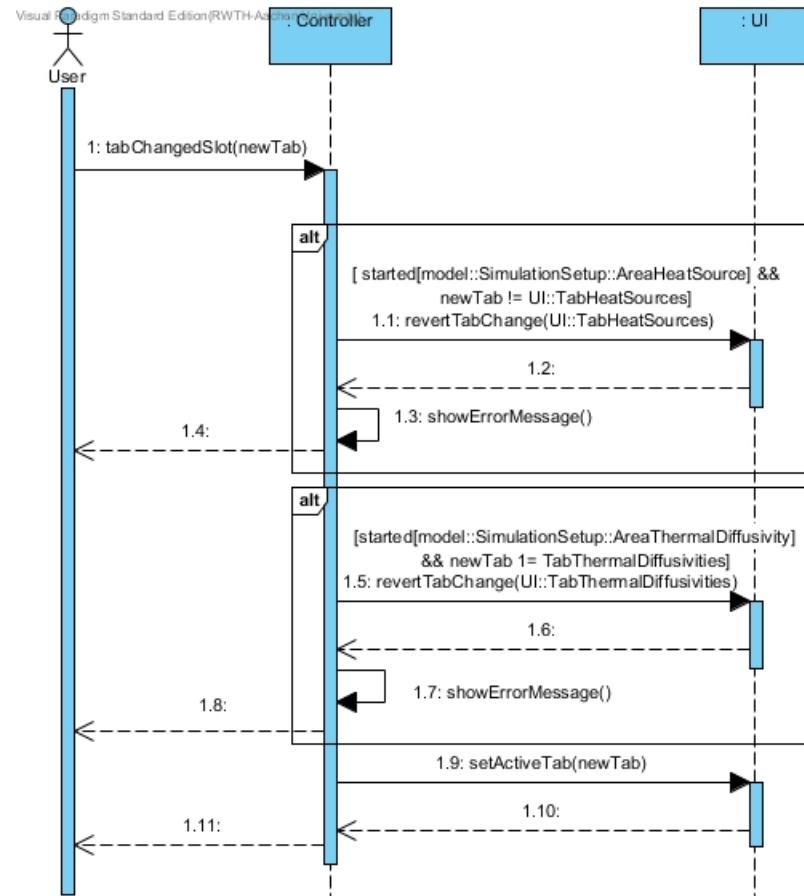


Abbildung 3.70: Sequenzdiagramm Controller::tabChangedSlot

## undoSlot

Das Sequenzdiagramm für *undoSlot* ist in Abbildung 3.71 dargestellt. *undoSlot* macht den letzten Mausklick auf eine der Gebiet-Platten rückgängig.

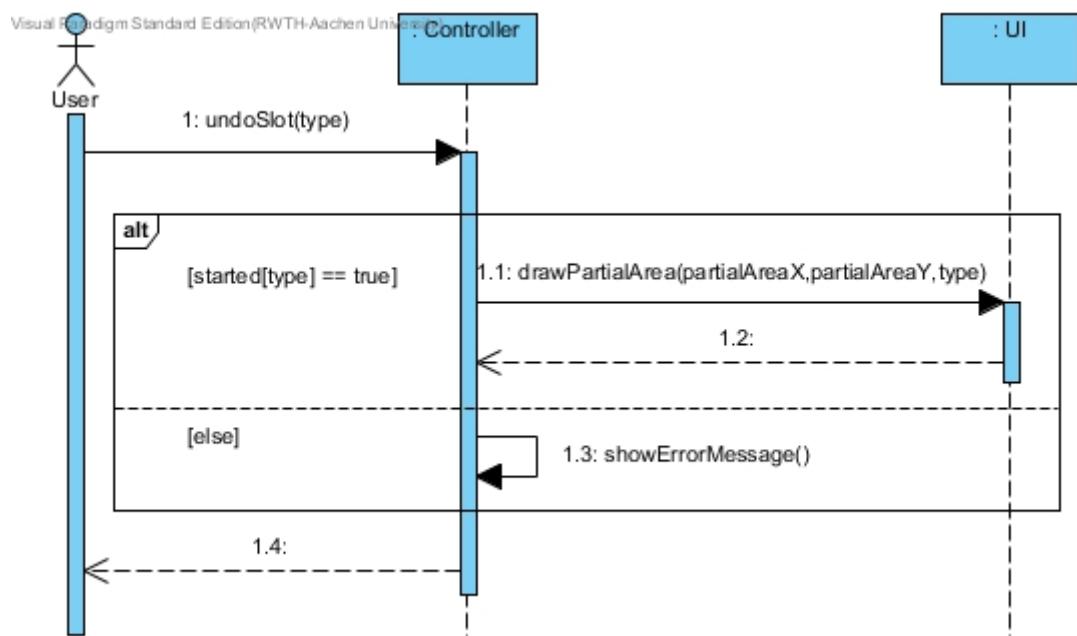


Abbildung 3.71: Sequenzdiagramm Controller::undoSlot

## useHeatSourcesSlot

Das Sequenzdiagramm für *useHeatSourcesSlot* ist in Abbildung 3.72 dargestellt. *useHeatSourcesSlot* verwaltet das Aktivieren der Nutzung der Wärmequellen-Gebiete für die Optimierung.

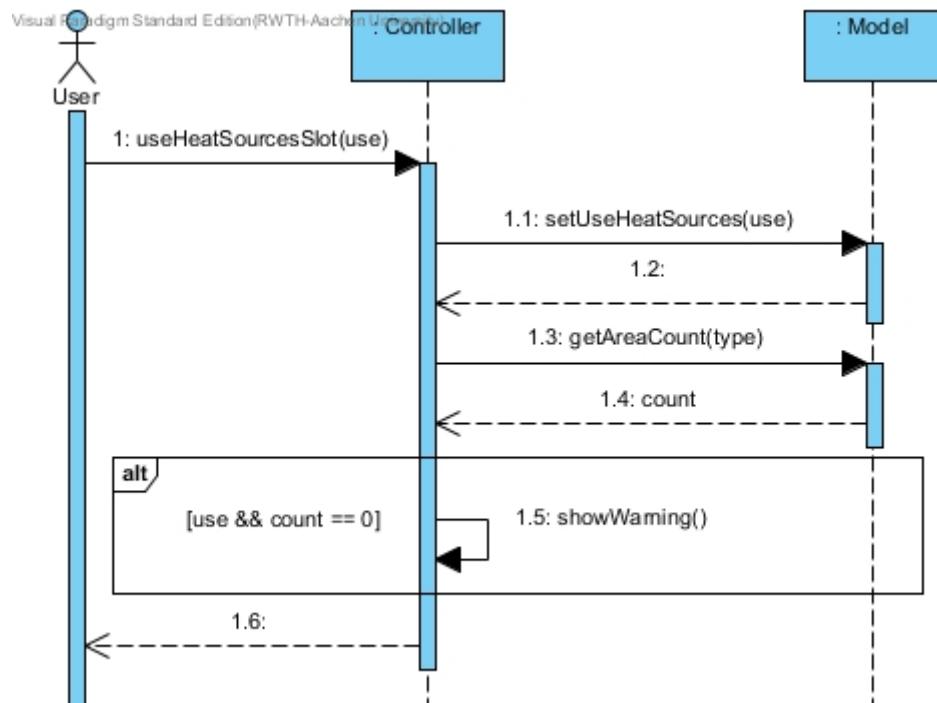


Abbildung 3.72: Sequenzdiagramm Controller::useHeatSourcesSlot

## visualizeStateSlot

Das Sequenzdiagramm für *visualizeStateSlot* ist in Abbildung 3.73 dargestellt. *selectIntMethodSlot* visualisiert einen einzelnen Zeitschritt der letzten Simulation.

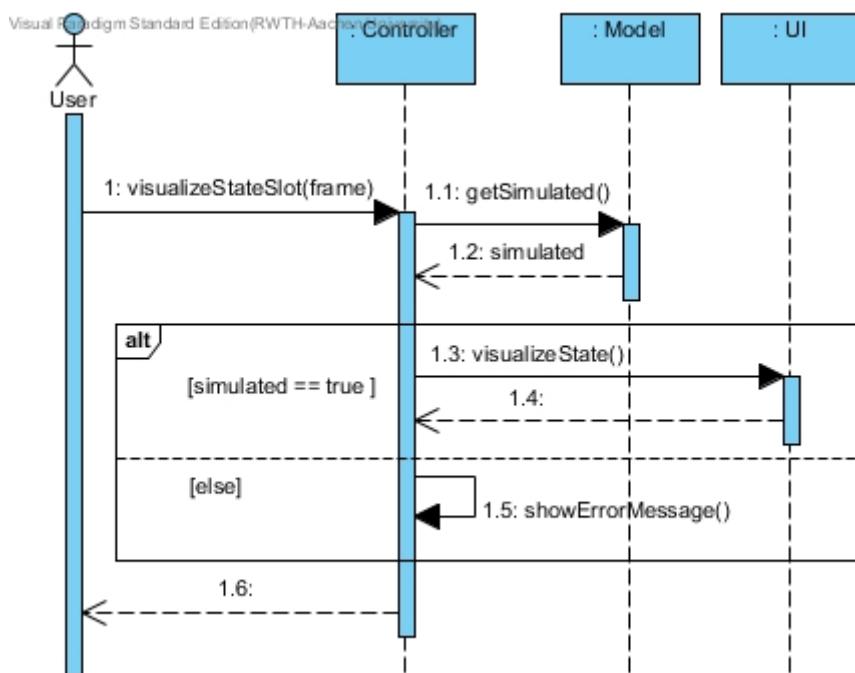


Abbildung 3.73: Sequenzdiagramm Controller::visualizeStateSlot

### 3.2.3.3 UI

#### drawPartialArea

Das Sequenzdiagramm für *drawPartialArea* ist in Abbildung 3.74 dargestellt. *drawPartialArea* lässt den entsprechenden Tab ein angefangenes Gebiet zeichnen.

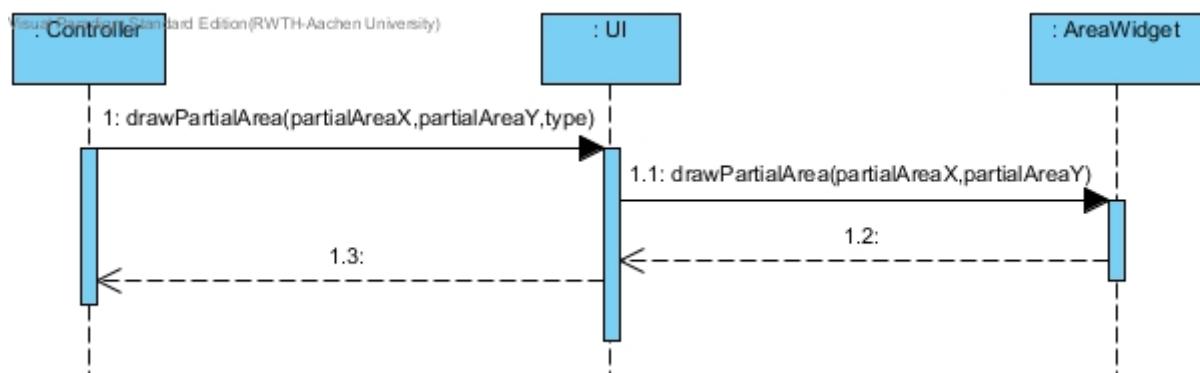


Abbildung 3.74: Sequenzdiagramm UI::drawPartialArea

## setController

Das Sequenzdiagramm für *setController* ist in Abbildung 3.75 dargestellt. *setController* setzt die Referenz auf den Controller und verbindet Signale und Slots.

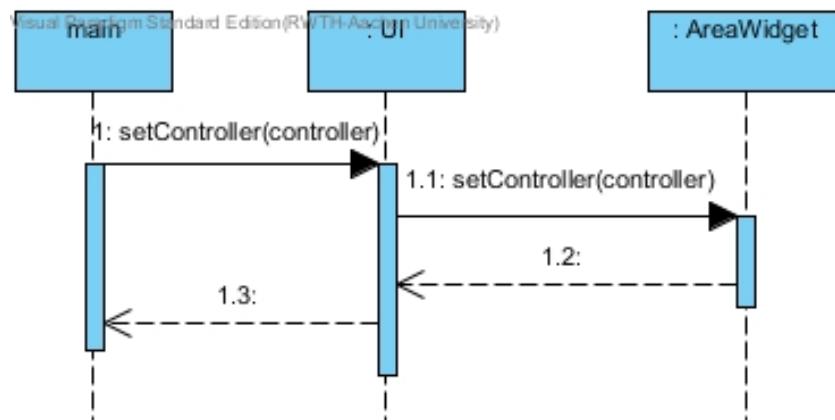


Abbildung 3.75: Sequenzdiagramm UI::setController

## setModel

Das Sequenzdiagramm für *setModel* ist in Abbildung 3.76 dargestellt. *setModel* setzt die Referenz auf das Modell und lädt einige Einstellungen.

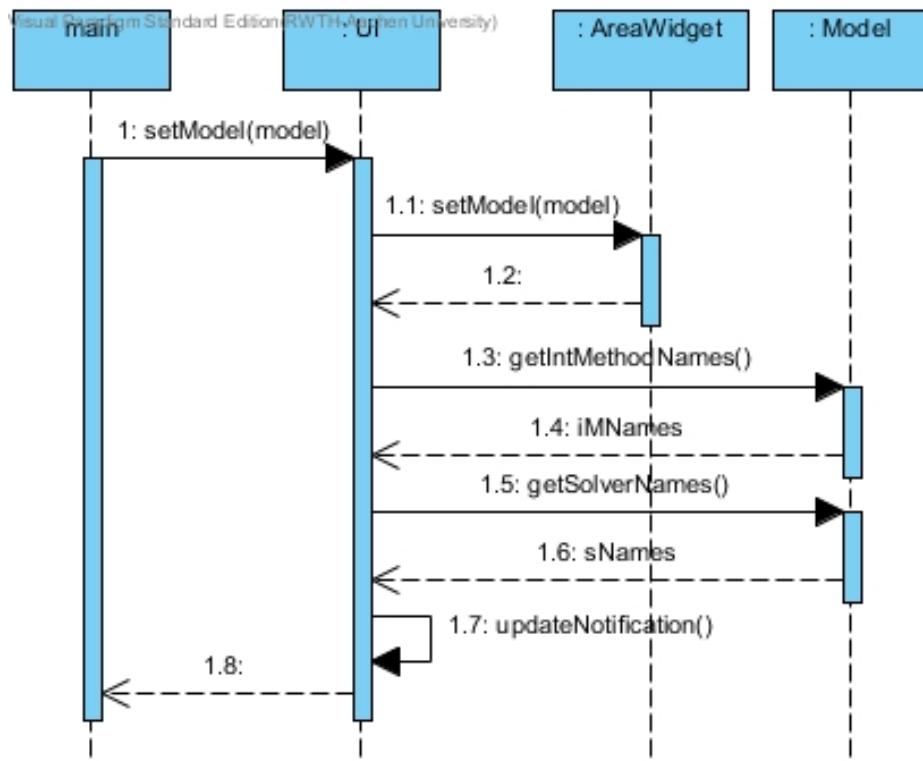


Abbildung 3.76: Sequenzdiagramm UI::setModel

## updateIBVs

Das Sequenzdiagramm für *updateIBVs* ist in Abbildung 3.77 dargestellt. *updateIBVs* updatet den Tab für die Rand- und den Anfangswert.

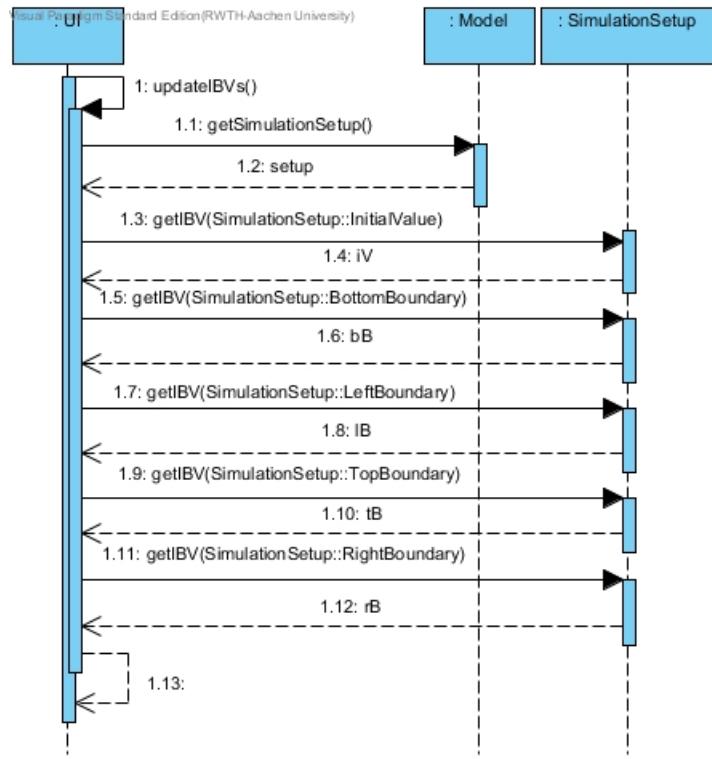


Abbildung 3.77: Sequenzdiagramm UI::updateIBVs

### updateNotification

Das Sequenzdiagramm für *updateNotification* ist in Abbildung 3.78 dargestellt. *updateNotification* updatet den aktuell aktiven (sichtbaren) Tab des UI.

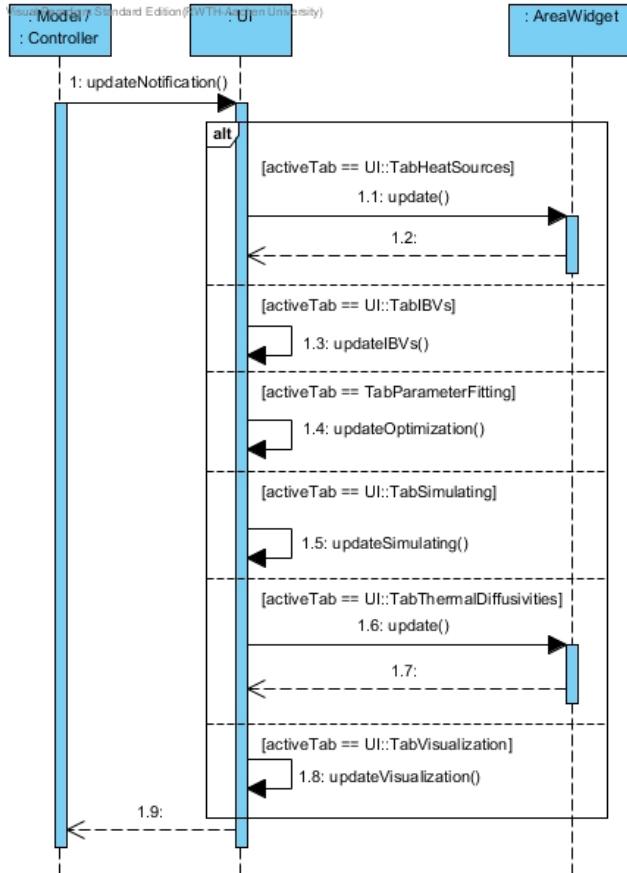


Abbildung 3.78: Sequenzdiagramm UI::updateNotification

## updateOptimization

Das Sequenzdiagramm für *updateOptimization* ist in Abbildung 3.79 dargestellt. *updateOptimization* updatet den Tab für das Durchführen einer Optimierung.

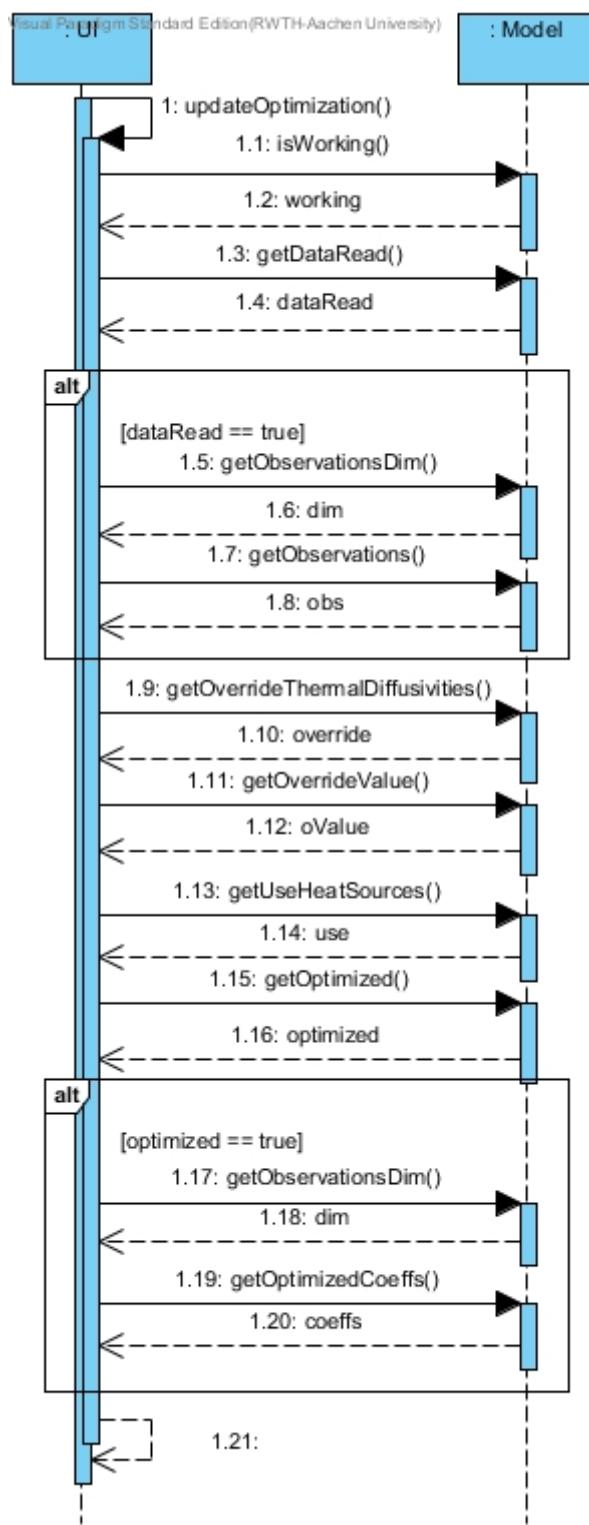


Abbildung 3.79: Sequenzdiagramm UI::updateOptimization

## updateSimulating

Das Sequenzdiagramm für *updateSimulating* ist in Abbildung 3.80 dargestellt. *updateSimulating* updatet den Tab für das Durchführen einer Simulation.

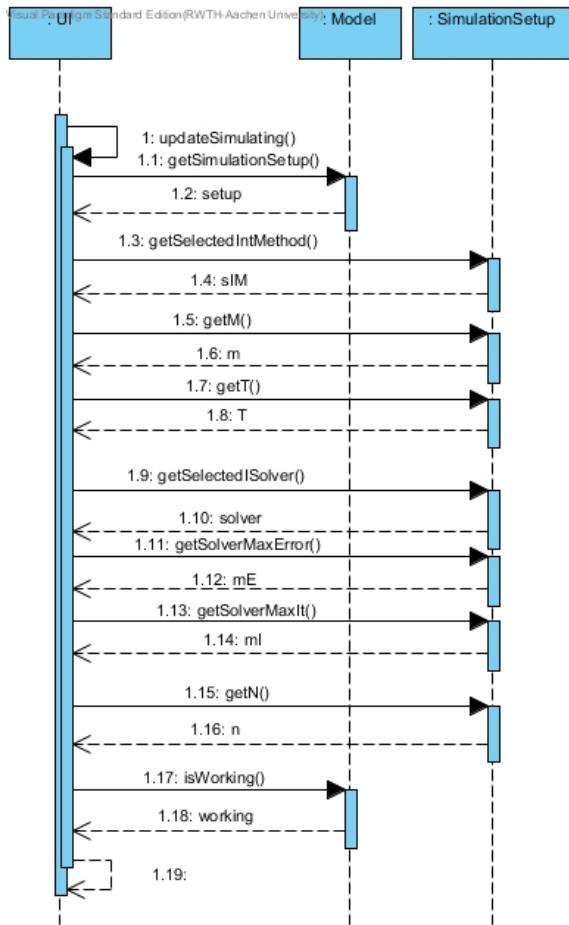


Abbildung 3.80: Sequenzdiagramm `UI::updateSimulating`

## updateVisualization

Das Sequenzdiagramm für `updateVisualization` ist in Abbildung 3.81 dargestellt. `updateIBVs` updatet den Tab für die Visualisierung von Simulationsergebnissen.

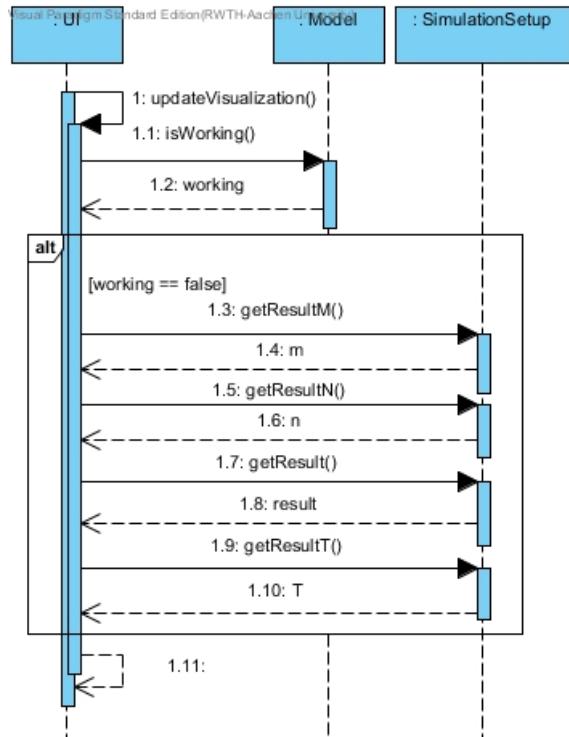


Abbildung 3.81: Sequenzdiagramm `UI::updateVisualization`

# Kapitel 4

## Benutzerdokumentation

### 4.1 Installation

Die Software wird als SEPGruppe11.zip ausgeliefert. Der erste Schritt ist die zip-Datei zu entpacken. Nach dem Entpacken befinden sich im Zielordner ein Unterordner *Quellcode* und diese Dokument. Die Voraussetzung, dass die Software installiert werden kann, ist ein QT5 fähiger Compiler. Um die Doxygen-Dokumentation erstellen zu können, muss Doxygen installiert sein.

#### 4.1.1 Linux

Zur Installation unter Windows muss im Ordner *Quellcode* das Projekt zunächst mit QMake bearbeitet werden, auf dem RWTH-Cluster lautet der entsprechenden Befehl hierzu *qmake-qt5*. Danach muss dann das Projekt mit dem Befehl *make* kompiliert werden und danach kann es mit dem Befehl *make run* ausgeführt werden. Alternativ kann auch direkt der Befehl *make run* ausgeführt werden, dann wird das Projekt kompiliert und direkt ausgeführt. Mit dem Befehl *make clean* kann das Programm wieder deinstalliert werden.

#### 4.1.2 Windows

Zur Installation unter Windows muss die *SEP\_WLeitung.pro* Datei im Ordner *Quellcode* über Qt-creator geöffnet werden und dort kompiliert werden.

#### 4.1.3 Doxygen

##### 4.1.3.1 Linux

Zum Erstellen der Doxygen-Dokumentation muss in den Unterordner *Quellcode Doxygen* gewechselt werden und dort shell-Befehl *doxygen doxygen-config* ausgeführt werden. Voraussetzung ist, dass Doxygen installiert und für die Nutzung per Konsole konfiguriert wurde.

##### 4.1.3.2 Windows

Zum Erstellen der Dokumentation muss im Doxy-Wizard die Konfigurationsdatei *doxygen-config* geladen werden und dann der *run*-Button betätigt werden.

## 4.2 Beispielsitzung

Bevor der Benutzer eine Simulation startet, kann er die folgenden Einstellungen festlegen, um ein Simulationsexperiment zu spezifizieren:

- Temperaturleitkoeffizienten
- Wärmequellen
- Anfangs- und Randwerte
- Diskretisierungsparameter

Nach Start des Programms wird die graphische Benutzeroberfläche angezeigt.

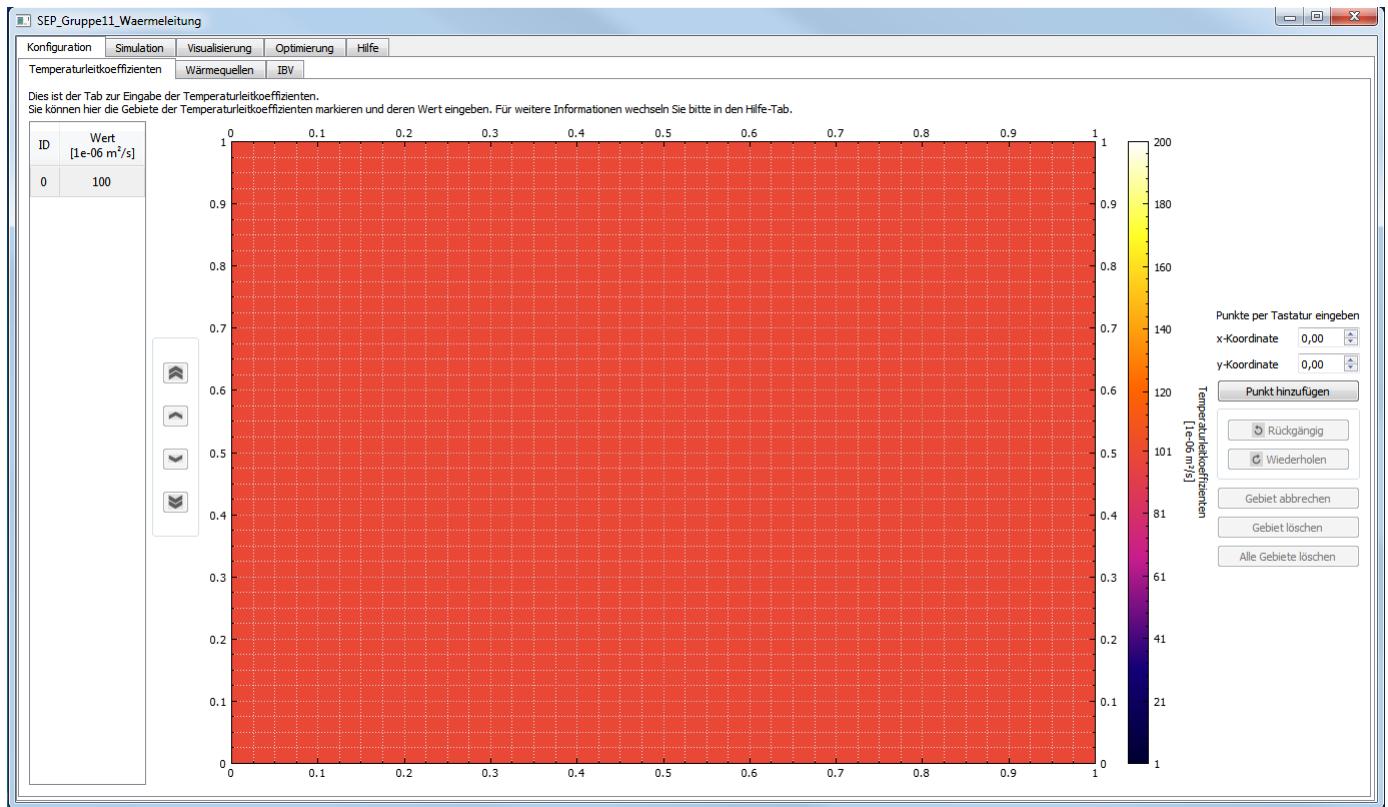


Abbildung 4.1: graphische Benutzeroberfläche

Im Temperaturleitkoeffizienten- und Wärmequellen-Tab kann der Benutzer Gebiete festlegen. In den beiden Tabs stehen dem Benutzer die folgenden Funktionalitäten zur Verfügung:

Neues Gebiet hinzufügen: Klicken Sie entweder mit der linken Maustaste an die gewünschte Stelle auf der Platte oder geben Sie die Koordinaten über die Felder 'x-Koordinate' und 'y-Koordinate' ein und bestätigen Sie mit 'Punkt hinzufügen'. Um die Eingabe eines Gebietes abzuschließen müssen Sie auf den Startpunkt des Gebietes klicken. Ein Gebiet ist nur dann korrekt, wenn es einfach wegzusammenhängend ist, d.h. Kanten dürfen sich nicht schneiden und es muss abgeschlossen sein. Wurde ein Gebiet korrekt eingegeben können Sie den Temperaturleitkoeffizienten (in  $1e-06 \text{ m}^2/\text{s}$ ) bzw den Wert der Wärmequelle zu diesem Gebiet eingeben.

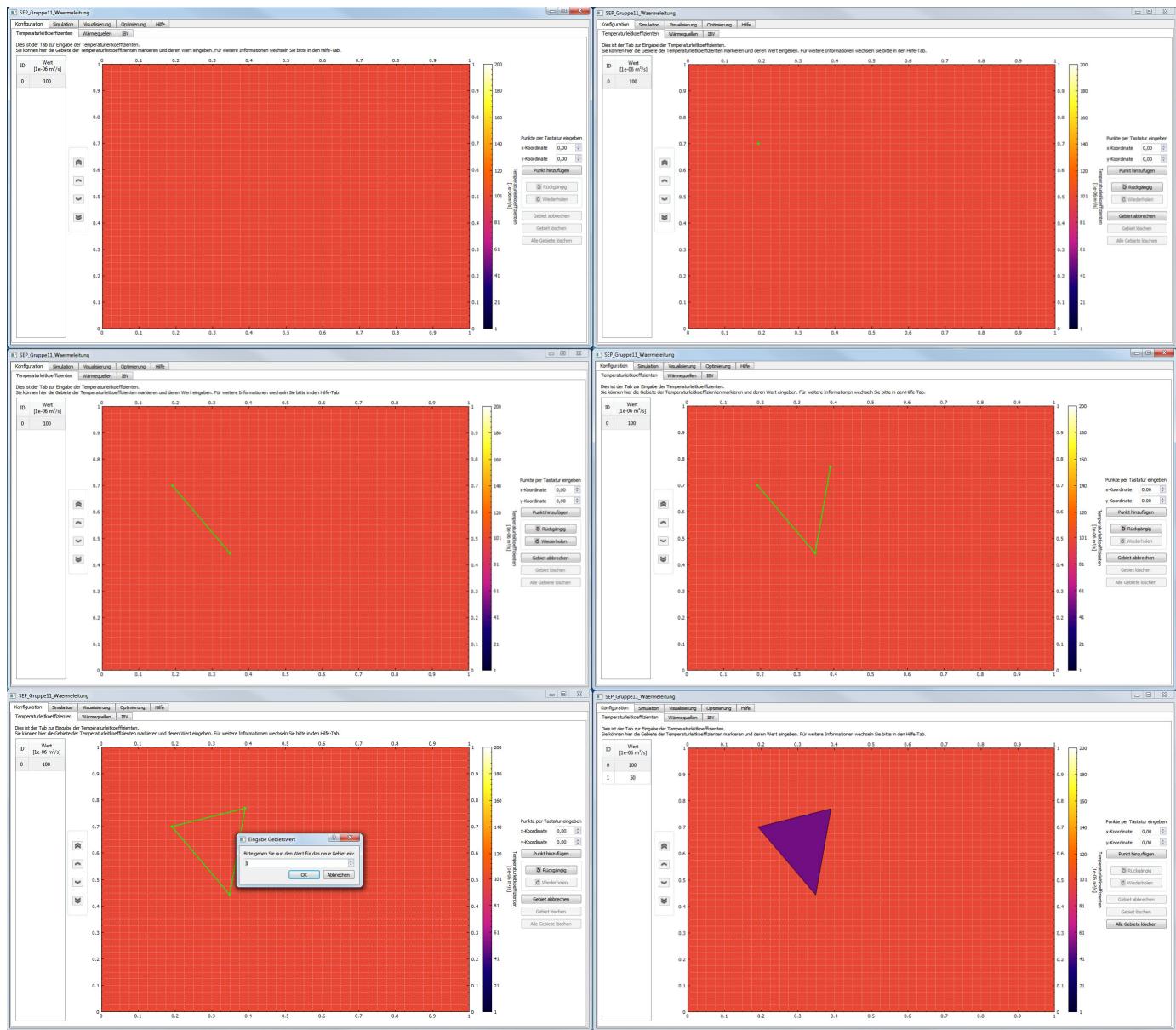


Abbildung 4.2: Gebiet hinzufügen

Einen Punkt rückgängig machen: Um während der Eingabe eines neuen Gebietes einen Punkt rückgängig zu machen, klicken Sie auf den Knopf 'Rückgängig'.

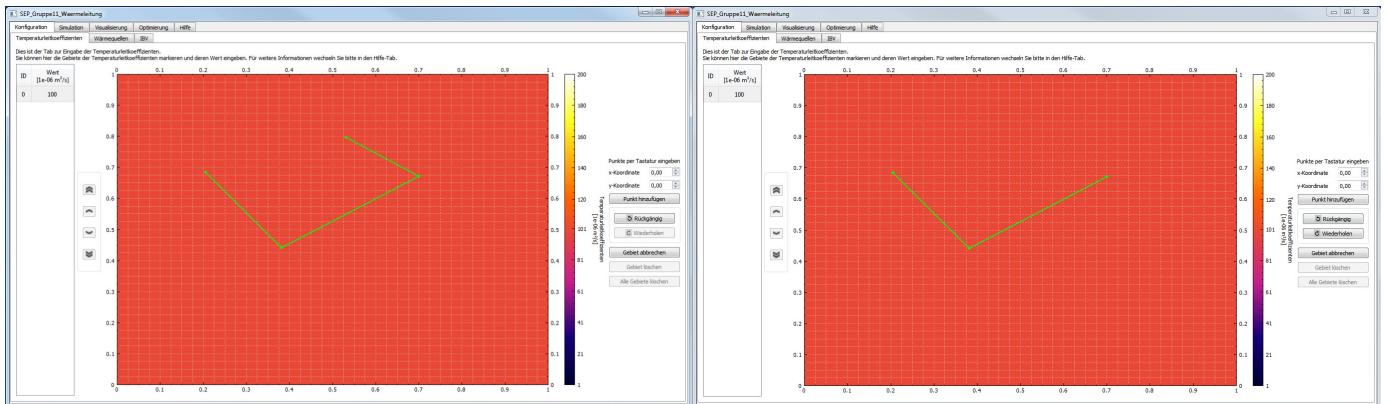


Abbildung 4.3: Einen Punkt rückgängig machen

Ein Gebiet abbrechen: Um das aktuell begonnene Gebiet zu löschen, klicken Sie auf den Knopf 'Gebiet abbrechen'.

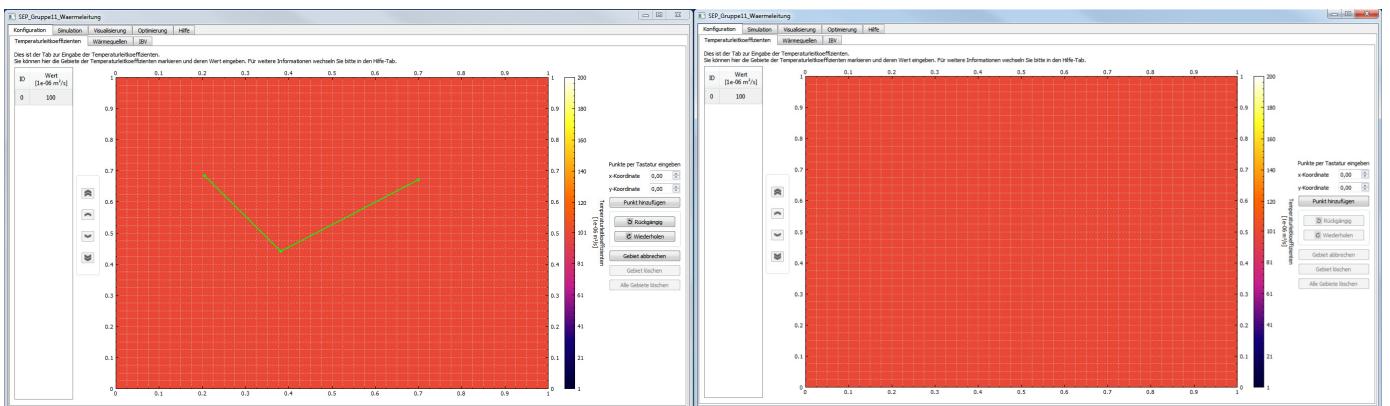


Abbildung 4.4: Ein Gebiet abbrechen

Ein Gebiet löschen: Um ein beliebiges Gebiet zu löschen muss der Radio-Button 'Gebiet auswählen', aktiviert sein. Wählen Sie nun auf der Platte das Gebiet aus, welches Sie löschen wollen und klicken Sie anschließend auf den Knopf 'Gebiet löschen'.

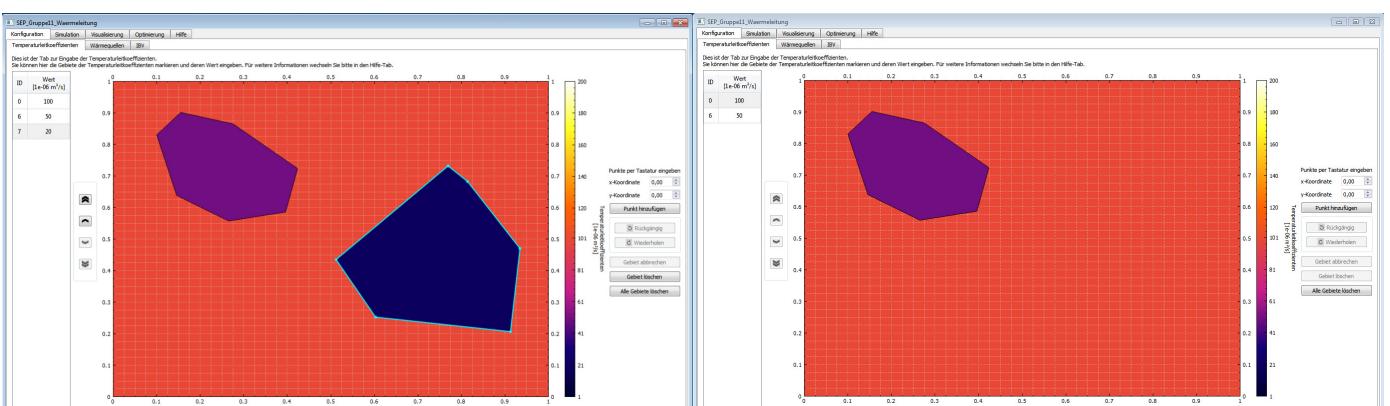


Abbildung 4.5: Ein Gebiet löschen

Alle Gebiete löschen: Um alle Gebiete zu löschen, klicken Sie auf den Knopf 'Alle Gebiete löschen'.

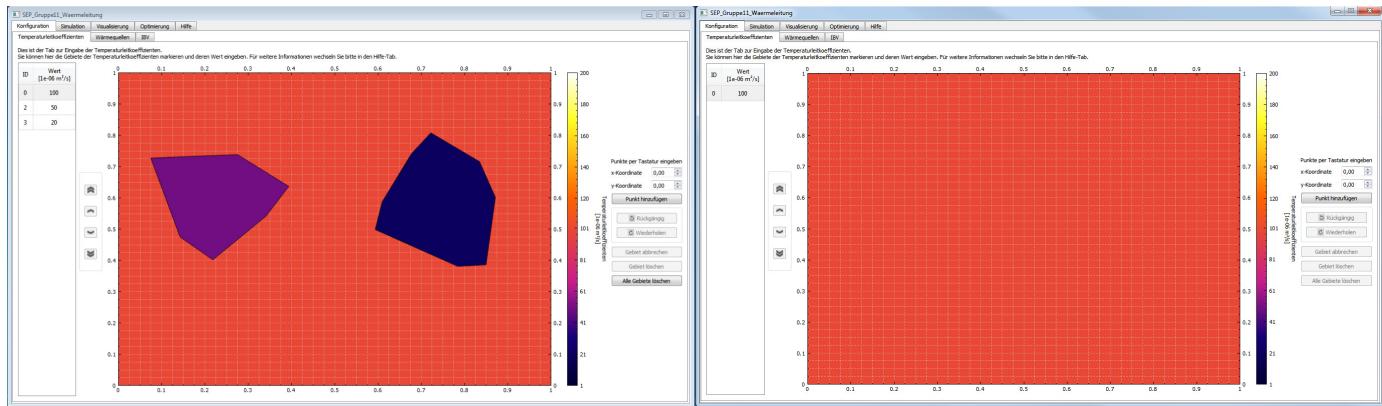


Abbildung 4.6: Alle Gebiete löschen

Reihenfolge der Gebiet tauschen: Sie können die Reihenfolge der eingegebenen Gebiete in der Tabelle ändern. Hierfür muss mehr als ein Gebiet auf der Platte eingegeben worden sein. Klicken Sie entweder mit der rechten Maustaste auf das gewünschte Gebiet auf der Platte oder klicken Sie auf die Zeile des entsprechenden Gebietes um dieses zur markieren. Anschließend können Sie die Reihenfolge mit den Pfeiltasten neben der Tabelle ändern. Mit einem Klick auf den einfachen Pfeil wird das ausgewählte Gebiet entweder um eine Position nach oben oder nach unten in der Tabelle verschoben. Mit einem Klick auf den doppelten Pfeil wird das ausgewählte Gebiet entweder an den Anfang oder ans Ende der Tabelle verschoben. Das Gebiet welches in der Tabelle an letzter Position steht, wird auf der Platte als oberstes Gebiet angezeigt.

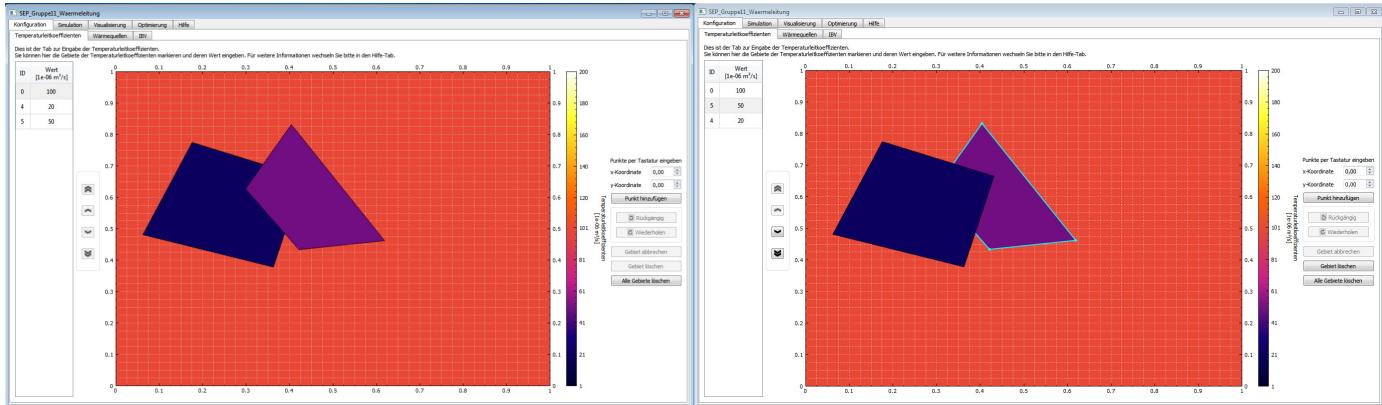


Abbildung 4.7: Reihenfolge ändern

Wert eines Gebietes ändern: Um den Temperaturleitkoeffizienten bzw den Wert der Wärmequelle eines Gebietes zu ändern, klicken Sie mit einem Doppelklick auf den Wert in der Tabelle in der Spalte 'Werte'. Anschließend wird das ausgewählte Gebiet auf der Platte markiert und Sie können den neuen Wert eingeben.

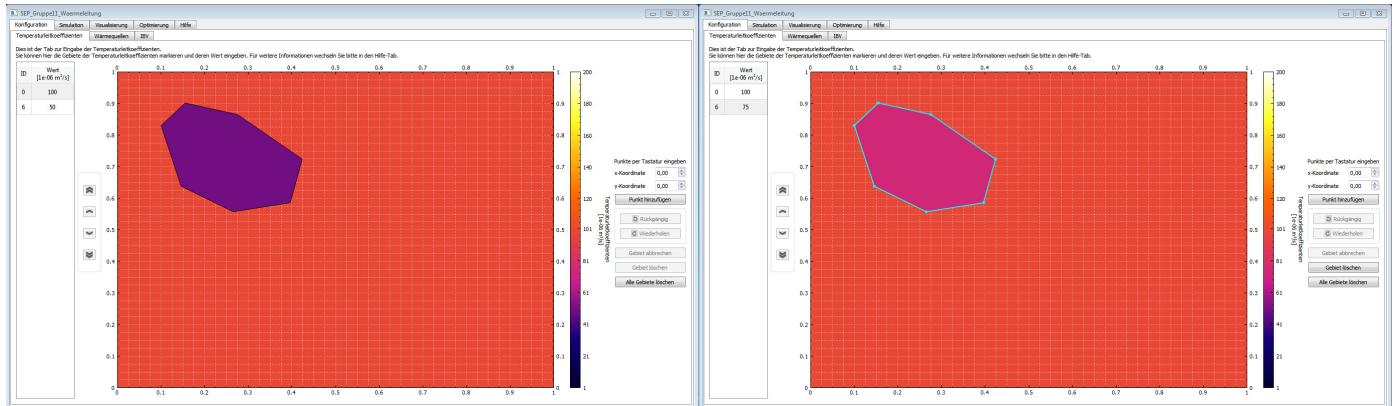


Abbildung 4.8: Wert eines Gebiets ändern

Der dritte Tab ist zur Einstellung der Anfangs- und Randwerte.

Der IBV-Tab bietet Ihnen folgende Funktionalitäten:

Um die Anfangswerte und Randwerte zu ändern, ändern Sie den gewünschten Wert in den Feldern 'Anfangswert eingeben', 'unteren Randwert eingeben', 'linken Randwert eingeben', 'rechten Randwert eingeben' oder 'oberen Randwert eingeben'.

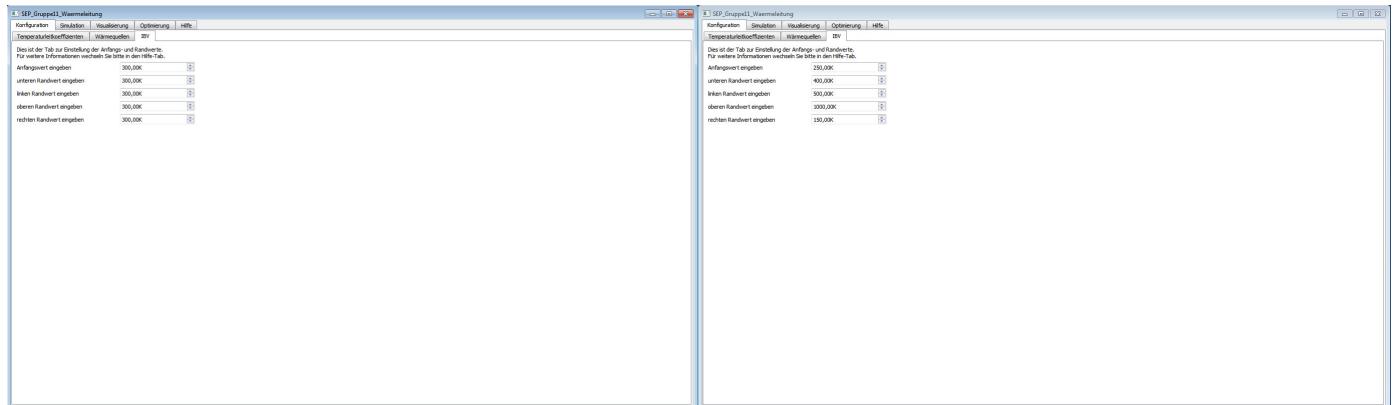


Abbildung 4.9: Anfangs- und Randwerte ändern

Der Simulations-Tab bietet Ihnen folgende Funktionalitäten:

1. Zeitdiskretisierung ändern: In der Box 'Zeitdiskretisierung' können Sie die Integrationsmethode, die Zeitschritte M und den Endzeitpunkt T ändern.
2. Linearen Gleichungssystemlöser einstellen: In der Box 'LGS Löser' können Sie den Löser, die relative Genauigkeit und die maximale Iterationszahl ändern.
3. Ortsdiskretisierung einstellen: In der Box 'Ortsdiskretisierung' können Sie die Stützstellen N eingeben und die Simulation starten, indem Sie auf den Knopf 'Simulieren' drücken. Außerdem können Sie eine gestartete Simulation abbrechen, hier wird das Ergebnis der vorherigen Simulation ungültig.
4. Simulationseinstellungen: In der Box 'Simulationseinstellungen' können Sie durch Klick auf den Knopf 'Speichern' alle gesetzten Einstellungen speichern. Ebenso können Sie zuvor gesetzte Einstellungen Laden, indem Sie auf den Knopf 'Laden' klicken. Indem Sie auf den Knopf 'Zurücksetzen' klicken, werden alle zuvor gemachten Einstellungen auf den Anfangszustand zurückgesetzt.

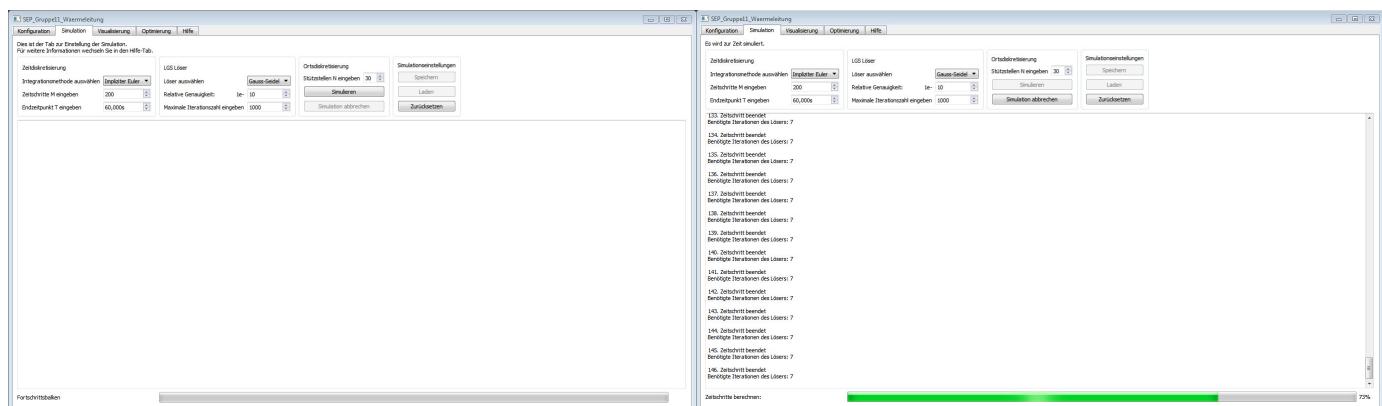


Abbildung 4.10: Start einer beliebigen Simulation

Der Visualisierungs-Tab bietet Ihnen folgende Funktionalitäten:

Nachdem eine Simulation erfolgreich angeschlossen worden ist, können Sie sich die Visualisierung der Temperaturverteilung als Video oder als Einzelbild anschauen.

1. Video starten: Um das Video zu starten, drücken Sie auf den Knopf 'Play'.
2. Bild anzeigen: Um ein Einzelbild anzeigen zu lassen, bewegen Sie den Schieberegler unter der Platte an die gewünschte Position.
3. Video versetzt anfangen lassen: Bewegen Sie den Schieberegler an die Position von der Sie das Video starten möchten und klicken Sie anschließend auf den Knopf 'Play'.

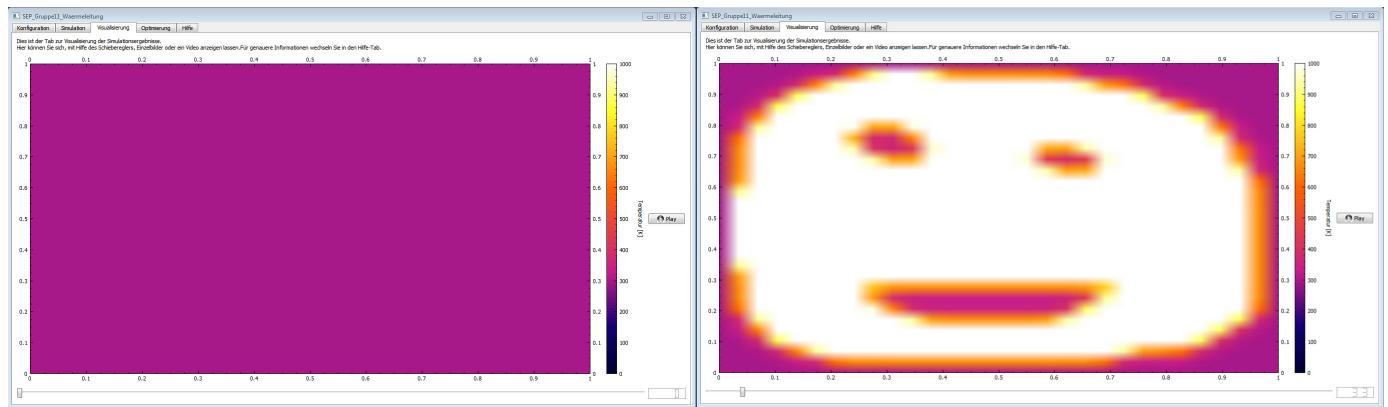


Abbildung 4.11: Visualisierung eines beliebigen Zustands

Der Optimierungs-Tab besteht aus zwei weiteren Tabs:

Der Konfigurations-Tab bietet Ihnen folgende Funktionalitäten:

1. Laden von Messungen/Beobachtungen: Indem Sie auf den Knopf 'Laden' klicken, können Sie Messungen/Beobachtungen einlesen lassen.
2. Vorhandene Wärmequellen nutzen: Indem Sie die Checkbox 'Nutze bereits vorhandene Wärmequellen zur Simulation' aktivieren, werden die eingegebenen Wärmequellen aus dem Tab Wärmequellen für die Simulation übernommen.
3. Anfangs Temperaturleitkoeffizienten ändern: Um die Anfangs Temperaturleitkoeffizienten zu ändern, aktivieren Sie die Checkbox 'Überschreibe bereits vorhandene Temperaturleitkoeffizienten zur Simulation' und ändern Sie anschließend den Wert im entsprechenden Feld.
4. Simulationseinstellungen ändern: Um die Simulationseinstellungen zu ändern wechseln Sie in den Simulations-Tab und ändern Sie dort die gewünschten Werte.
5. Optimierung starten: Indem Sie auf den Knopf 'Optimierung starten' klicken, wird die Optimierung gestartet und anhand eines Fortschrittsbalken wird der Fortschritt angezeigt.

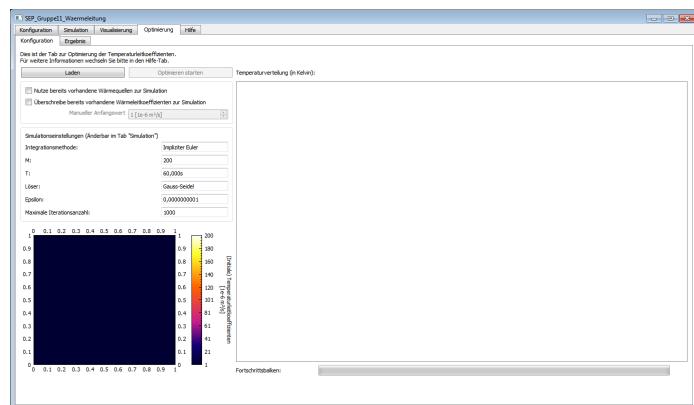


Abbildung 4.12: Optimierung starten

Der Ergebnis-Tab zeigt die gefitteten Temperaturleitkoeffizienten in Form einer Tabelle an.

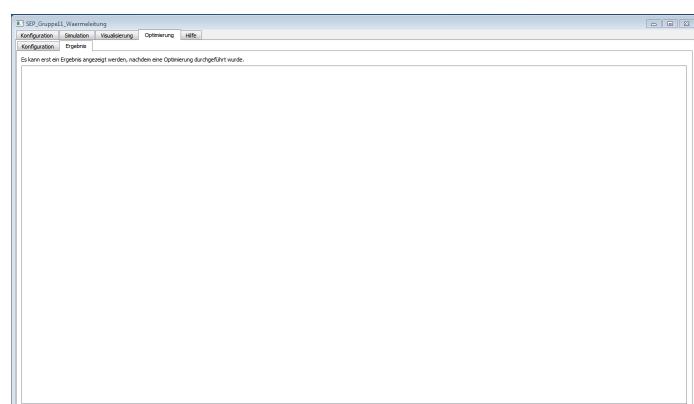


Abbildung 4.13: gefitteten Temperaturleitkoeffizienten

## 4.3 Fehlersituationen

Während der Benutzer ein Gebiet eingibt, darf er den Tab nicht wechseln.

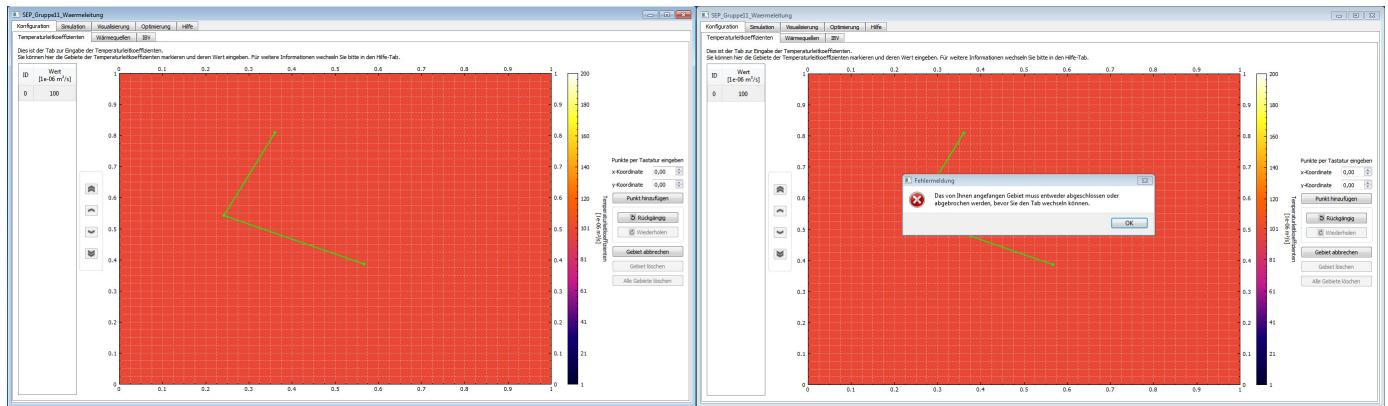


Abbildung 4.14: Fehler Tab Wechseln

Ein korrektes Gebiet muss einfach wegzusammenhängend sein, d.h. Kanten dürfen sich nicht schneiden.

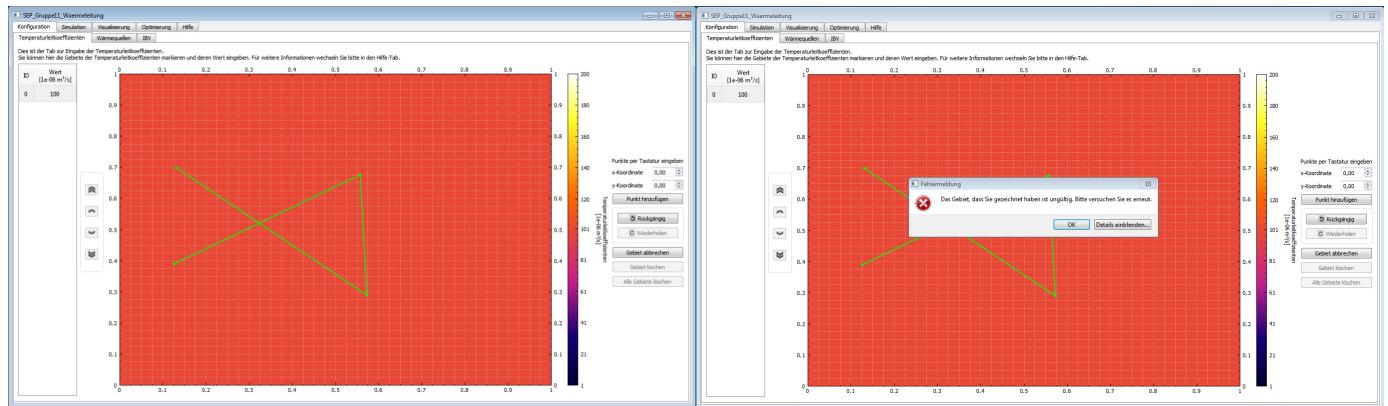


Abbildung 4.15: Fehler Gebiet eingeben

# Kapitel 5

## Entwicklerdokumentation

### 5.1 Codestruktur

Das Programm wurde im Stil eines Model-View-Controller-Architekturmusters entwickelt. Daher besteht es vor allem aus den drei Hauptklassen *Model*, *Controller* und *UI*.

Die Klasse *UI* implementiert das User Interface und stellt jeweils abhängig vom aktiven Tab einen gewissen Teil des Modells dar. Von diesem UI gelangen die Benutzerinteraktionen in Form von Qt-Signalen an den Controller, der diese verarbeitet, d.h. insbesondere z.B. Werte auf Gültigkeit überprüft, und dann entweder im Erfolgsfall an das Modell weiterleitet oder im Fehlerfall das UI informiert, sodass sich dieses zurücksetzt. Das Modell sendet immer sobald sich etwas an seinem Zustand ändert, d.h. insbesondere sich Simulationseinstellungen ändern eine Update-Nachricht nach dem Beobachter-Entwurfsmuster an das UI, worauf dieses die angezeigten Informationen aus dem Modell aktualisiert. Des Weiteren enthält das Modell die „Modell-Logik“, d.h. die Funktionalität zum Durchführen von Simulationen der Wärmeleitungsgleichung und von Optimierungen der Temperaturleitkoeffizienten.

Die Aufteilung der Klassen in Pakete ist in Kapitel 3 dargestellt, das Paket *presentation* enthält die *Controller*-Klasse und die *UI*- und alle dazugehörigen Widget-Klassen, das Paket *model* enthält die *Model*- und dazugehörige Klassen und das Paket *algorithms* enthält alle mathematischen/numerischen Algorithmen, d.h. insbesondere die Integrationsmethoden und LGS Löser.

### 5.2 Detaillierte Dokumentation des Codes

Kurzbeschreibungen der Funktionen aller Klassen finden Sie in Kapitel 3.2 bei den Sequenzdiagrammen der jeweiligen Funktion. Eine ausführlichere Dokumentation der ganzen Klassen ist in der Doxygen-Dokumentation enthalten, in Kapitel 4.1.3 wird erklärt wie diese erstellt werden kann.

### 5.3 Software Tests

Zur Verifikation der Software haben wir verschiedene Testmethoden verwendet. Im Folgenden werden diese Tests nach Code-Zugehörigkeit sortiert aufgelistet. Im mitgelieferten Projekt *Testklassen* sind ein Teil der im folgenden genannten Tests implementiert.

#### 5.3.1 presentation

Die Funktionalität des UI und des Controllers, d.h. insbesondere die Benutzerinteraktion, wurde durch manuelle Use-Case-basierte Tests nach den Use Case von 2.1.2 getestet.

### 5.3.2 model

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 A.1 präsentiert. Die weiteren numerischen Methoden wurden anschließend gegen dieses Referenzergebnis getestet.

#### 5.3.2.1 Area

Die Funktionen *insidePoint* und *onLine* und dadurch von diesen benutzte Funktionen wurden durch eine Robustness-Worst-Case-Teststrategie (nach der Vorlesung „Software Engineering“) mit zufällig generierten Geraden bzw. Quadranten. Des Weiteren wurde die Funktion *segIntersect* mit einem weiterem speziellen Fall getestet, der vom ursprünglichem Algorithmus nicht abgedeckt wurde.

### 5.3.3 algorithms

#### 5.3.3.1 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

#### 5.3.3.2 Solver

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

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

# Anhang A

## Anhang

### A.1 Matlab-Code

Listing A.1: HeatImpEuler.m

```
function [t,u,steps] = HeatImpEuler(u0,T,m,A,g,f)
% Liefert Approximation (impliziter Euler) der Loesung der durch A orts-
    diskretisierten PDE
% und Vektor mit berechneten Zeitpunkten (t)
dt = T/m;
Nx = size(A,1);
n = sqrt(Nx);
dx = 1/(n-1);

e = ones(Nx,1);
B = diag(e) - dt/(dx*dx) * A;

% Approximation fuer das gewuenschte Zeitintervall
t = zeros(m+1,1);
t(1) = 0;
u = zeros(Nx,m+1);
u(:,1) = u0;

for i=2:m+1
    [u(:,i),steps(i,1)] = JC2(B,(u(:,i-1)+dt*f));
    for j=1:n
        u(j,i) = g(i);
        u(n*n-j+1,i) = g(4*n-j+1);
    end
    for j=1:n
        u((j-1)*n+1,i) = g(n+2*j-1);
        u(j*n,i) = g(n+2*j);
    end
    t(i) = t(1) + (i-1) * dt;
end

end
```

Listing A.2: JC.m

```
function [u,steps] = JC(A,b)
% iterativer Loeser: Jacobi-Verfahren

maxIt = 100;
N = size(A,1); % Dimension des Problems
eps = 1e-5; % Abbruchkriterium
```

```

steps = 0; % Anzahl Iterationen
old = zeros(N,1); % Startvektor
res = norm(b-A*old,2); % r0 in der unendlich-Norm

while( res - eps > 0 && steps < maxIt)
    steps = steps+1;
    for i=1:N
        sum = 0;
        for j=1:i-1
            sum = sum + A(i,j) * old(j);
        end
        for j=i+1:N
            sum = sum + A(i,j) * old(j);
        end
        u(i,1) = 1/A(i,i) * (b(i) - sum);
    end
    old = u;
    res = norm(b-A*u,2); % rSteps
end

end

```

Listing A.3: SEP.m

```

clc
clear all

m = 2;
n = 5; deltaX = 1/(n-1);
T = 1; deltaT = 1/m;

u0 = zeros(n*n,1); g = zeros(4*n,1); f = zeros(n*n,1);

for i=1:n
    g(i) = 300;
    g(4*n+1-i) = 300;
end
for i=1:n
    g(n+2*i-1) = 300;
    g(n+2*i) = 300;
end

for i=1:(n*n)
    u0(i) = 100;
end
for i=1:n
    u0(i) = g(i);
    u0(n*n-i+1) = g(4*n-i+1);
end
for i=1:n
    u0((i-1)*n+1) = g(n+2*i-1);
    u0(i*n) = g(n+2*i);
end

for i=1:(n*n)
    f(i) = 10; % -32
end
for i=1:n
    f(i) = 0;
    f(n*n-i+1) = 0;
end
for i=1:n

```

```

f((i-1)*n+1) = 0;
f(i*n) = 0;
end

U = reshape(u0,[n,n]);

A = zeros(n*n,n*n);

for i=2:n-1
    for j=2:n-1
        A((i-1)*n+j,(i-2)*n+j) = 1;
        A((i-1)*n+j,(i-2)*n+j+n-1) = 1;
        A((i-1)*n+j,(i-2)*n+j+n) = -4;
        A((i-1)*n+j,(i-2)*n+j+n+1) = 1;
        A((i-1)*n+j,(i-2)*n+j+2*n) = 1;
    end
end

tC = zeros(n*n,1);
for i=1:n*n
    tC(i) = 5;
end
A = diag(tC) * A;

[t,u,steps] = HeatImpEuler(u0,T,m,A,g,f);
for i=1:size(u,2)
    U(:,:,i) = reshape(u(:,:,i),[n,n]);
end
%U
%surf(0:deltaX:1,0:deltaX:1,U(:,:,m))

```

# Abbildungsverzeichnis

2.1	Anwendungsfalldiagramm . . . . .	4
2.2	modifiziertes Anwendungsfalldiagramm . . . . .	5
2.3	Aktivitätsdiagramm Use Case Anfangsbedingungen eingeben . . . . .	12
2.4	Aktivitätsdiagramm Use Case Diskretisierungsgrößen eingeben . . . . .	12
2.5	Aktivitätsdiagramm Use Case Messungen eingeben . . . . .	12
2.6	Aktivitätsdiagramm Use Case Messungen importieren . . . . .	13
2.7	Aktivitätsdiagramm Use Case Randbedingungen eingeben . . . . .	13
2.8	Aktivitätsdiagramm Use Case Parameter optimieren . . . . .	13
2.9	Aktivitätsdiagramm Use Case Simulationsexperiment laden/speichern . . . . .	14
2.10	Aktivitätsdiagramm Use Case Simulieren . . . . .	14
2.11	Aktivitätsdiagramm Use Case Temperaturleitkoeffizienten eingeben . . . . .	15
2.12	Aktivitätsdiagramm Use Case Video abspielen . . . . .	15
2.13	Aktivitätsdiagramm Use Case Wärmequellen eingeben . . . . .	16
2.14	Aktivitätsdiagramm Use Case Zustand anzeigen . . . . .	16
2.15	Begriffsnetz . . . . .	18
3.1	Paketstruktur . . . . .	19
3.2	Klassendiagramm algorithms . . . . .	20
3.3	Sequenzdiagramm IntMethod::calcNextStep . . . . .	21
3.4	Sequenzdiagramm setUpSpecific . . . . .	21
3.5	Klassendiagramm Model . . . . .	22
3.6	Sequenzdiagramm Model::abortWork . . . . .	23
3.7	Sequenzdiagramm Model::addNewArea . . . . .	23
3.8	Sequenzdiagramm Model::deleteArea . . . . .	24
3.9	Sequenzdiagramm Model::finishedOptimizationSlot . . . . .	24
3.10	Sequenzdiagramm Model::finishedReadingDataSlot . . . . .	24
3.11	Sequenzdiagramm Model::finishedSimulationSlot . . . . .	25
3.12	Sequenzdiagramm Model::getObservations . . . . .	25
3.13	Sequenzdiagramm Model::getResult . . . . .	26
3.14	Sequenzdiagramm Model::getResultM . . . . .	26
3.15	Sequenzdiagramm Model::getResultN . . . . .	27
3.16	Sequenzdiagramm Model::getResultT . . . . .	27
3.17	Sequenzdiagramm Model::loadSetup . . . . .	28
3.18	Sequenzdiagramm Model::optimize . . . . .	28
3.19	Sequenzdiagramm Model::readObservations . . . . .	28
3.20	Sequenzdiagramm Model::removeLastArea . . . . .	29
3.21	Sequenzdiagramm Model::reorderArea . . . . .	29
3.22	Sequenzdiagramm Model::resetSetup . . . . .	30
3.23	Sequenzdiagramm Model::selectIntMethod . . . . .	30
3.24	Sequenzdiagramm Model::selectSolver . . . . .	30
3.25	Sequenzdiagramm Model::setAreaBackgroundColor . . . . .	31
3.26	Sequenzdiagramm Model::setIBV . . . . .	31
3.27	Sequenzdiagramm Model::setM . . . . .	32

3.28 Sequenzdiagramm Model::setN . . . . .	32
3.29 Sequenzdiagramm Model::setOverrideThermalDiffusivities . . . . .	32
3.30 Sequenzdiagramm Model::setOverrideValue . . . . .	33
3.31 Sequenzdiagramm Model::setSolverMaxError . . . . .	33
3.32 Sequenzdiagramm Model::setSolverMaxIt . . . . .	34
3.33 Sequenzdiagramm Model::setT . . . . .	34
3.34 Sequenzdiagramm Model::setUseHeatSources . . . . .	35
3.35 Sequenzdiagramm Model::simulate . . . . .	35
3.36 Sequenzdiagramm Model::updateAreaValue . . . . .	35
3.37 Sequenzdiagramm SimulationSetup::getContainingAreaID . . . . .	36
3.38 Sequenzdiagramm SimulationSetup::updateAreaValue . . . . .	36
3.39 Sequenzdiagramm SimulationWorker::simpleSimulation . . . . .	37
3.40 Sequenzdiagramm SimulationWorker::startOptimizationSlot . . . . .	38
3.41 Sequenzdiagramm SimulationWorker::startSimulationSlot . . . . .	39
3.42 Klassendiagramm presentation . . . . .	40
3.43 Sequenzdiagramm AreaWidget::drawPartialArea . . . . .	40
3.44 Sequenzdiagramm AreaWidget::update . . . . .	41
3.45 Sequenzdiagramm Controller::abortWorkSlot . . . . .	42
3.46 Sequenzdiagramm Controller::areaClickSlot . . . . .	42
3.47 Sequenzdiagramm Controller::areaValueChangedSlot . . . . .	43
3.48 Sequenzdiagramm Controller::clearAreasSlot . . . . .	43
3.49 Sequenzdiagramm Controller::deleteAreaSlot . . . . .	44
3.50 Sequenzdiagramm Controller::discardAreaSlot . . . . .	44
3.51 Sequenzdiagramm Controller::loadObservationsSlot . . . . .	45
3.52 Sequenzdiagramm Controller::loadSimulationSetupSlot . . . . .	45
3.53 Sequenzdiagramm Controller::newIBVValueSlot . . . . .	46
3.54 Sequenzdiagramm Controller::newMaxErrorSlot . . . . .	46
3.55 Sequenzdiagramm Controller::newMaxItSlot . . . . .	47
3.56 Sequenzdiagramm Controller::newMSlot . . . . .	47
3.57 Sequenzdiagramm Controller::newNSlot . . . . .	48
3.58 Sequenzdiagramm Controller::newOverrideValueSlot . . . . .	48
3.59 Sequenzdiagramm Controller::newTSlot . . . . .	49
3.60 Sequenzdiagramm Controller::optimizationSlot . . . . .	49
3.61 Sequenzdiagramm Controller::overrideThermalDiffusivitiesSlot . . . . .	50
3.62 Sequenzdiagramm Controller::playVideoSlot . . . . .	50
3.63 Sequenzdiagramm Controller::redoSlot . . . . .	51
3.64 Sequenzdiagramm Controller::reorderAreaSlot . . . . .	51
3.65 Sequenzdiagramm Controller::resetSimulationSetupSlot . . . . .	52
3.66 Sequenzdiagramm Controller::saveSimulationSetupSlot . . . . .	52
3.67 Sequenzdiagramm Controller::selectIntMethodSlot . . . . .	53
3.68 Sequenzdiagramm Controller::selectIterativeSolverSlot . . . . .	53
3.69 Sequenzdiagramm Controller::simulateSlot . . . . .	54
3.70 Sequenzdiagramm Controller::tabChangedSlot . . . . .	54
3.71 Sequenzdiagramm Controller::undoSlot . . . . .	55
3.72 Sequenzdiagramm Controller::useHeatSourcesSlot . . . . .	55
3.73 Sequenzdiagramm Controller::visualizeStateSlot . . . . .	56
3.74 Sequenzdiagramm UI::drawPartialArea . . . . .	56
3.75 Sequenzdiagramm UI::setController . . . . .	57
3.76 Sequenzdiagramm UI::setModel . . . . .	57
3.77 Sequenzdiagramm UI::updateIBVs . . . . .	58
3.78 Sequenzdiagramm UI::updateNotification . . . . .	58
3.79 Sequenzdiagramm UI::updateOptimization . . . . .	59

3.80 Sequenzdiagramm UI::updateSimulating . . . . .	60
3.81 Sequenzdiagramm UI::updateVisualization . . . . .	60
4.1 graphische Benutzeroberfläche . . . . .	62
4.2 Gebiet hinzufügen . . . . .	63
4.3 Einen Punkt rückgängig machen . . . . .	64
4.4 Ein Gebiet abbrechen . . . . .	64
4.5 Ein Gebiet löschen . . . . .	64
4.6 Alle Gebiete löschen . . . . .	65
4.7 Reihenfolge ändern . . . . .	65
4.8 Wert eines Gebiets ändern . . . . .	66
4.9 Anfangs- und Randwerte ändern . . . . .	66
4.10 Start einer beliebigen Simulation . . . . .	67
4.11 Visualisierung eines beliebigen Zustands . . . . .	68
4.12 Optimierung starten . . . . .	69
4.13 gefitteten Temperaturleitkoeffizienten . . . . .	69
4.14 Fehler Tab Wechseln . . . . .	70
4.15 Fehler Gebiet eingeben . . . . .	70

# Tabellenverzeichnis

2.1	Beschreibung Use Case Anfangsbedingungen eingeben . . . . .	6
2.2	Beschreibung Use Case Diskretisierungsgrößen eingeben . . . . .	6
2.3	Beschreibung Use Case Messungen eingeben . . . . .	7
2.4	Beschreibung Use Case Messungen importieren . . . . .	7
2.5	Beschreibung Use Case Randbedingungen eingeben . . . . .	8
2.6	Beschreibung Use Case Parameter optimieren . . . . .	8
2.7	Beschreibung Use Case Simulationsexperiment laden/speichern . . . . .	9
2.8	Beschreibung Use Case Simulieren . . . . .	9
2.9	Beschreibung Use Case Temperaturleitkoeffizienten eingeben . . . . .	10
2.10	Beschreibung Use Case Video abspielen . . . . .	10
2.11	Beschreibung Use Case Wärmequellen eingeben . . . . .	11
2.12	Beschreibung Use Case Zustand anzeigen . . . . .	11