

## ENTWICKLUNG EINER VIRTUAL REALITY-UMGEBUNG ZUR INTERAKTIVEN THERMISCHEN RAUMSIMULATION

C. Nytsch-Geusen<sup>1</sup> und K. Mathur<sup>1</sup>

<sup>1</sup>Institut für Architektur und Städtebau, UdK Berlin, Berlin, Deutschland

Email: nytsch@udk-berlin.de

### KURZFASSUNG

Im vorliegenden Beitrag wird ein Ansatz beschrieben, bei dem der Modellierungs- und Simulationsprozess für das Innenraumklima von Gebäuden vollständig in einer Virtual Reality (VR)-Umgebung stattfindet. Auf diese Weise wird der Simulationsexperte bezüglich des Raumdesigns, der räumlichen Wahrnehmung und der Interaktion mit dem Raummodell immersiv eingebunden. Das durch die Interaktionen sich ergebende simulierte Innenraumklima und weitere Größen wie z.B. Energiebedarfe werden dem Nutzer in der VR-Umgebung als Feedback visualisiert.

Zur softwaretechnischen Realisierung des beschriebenen Ansatzes wurde das Modelica-Werkzeug Dymola mit der Spiele-Engine Unity zu einem Gesamtsystem integriert, wobei zwischen beiden Programmen ein bidirektionaler Datenaustausch erfolgt. Die Basis der Berechnungen in Dymola bilden hierbei das thermische Gebäudemodell der Modelica-Bibliothek BuildingSystems. In Unity wurde zur Realisierung der VR-Simulationsumgebung ein Head-Mounted Display eingebunden und die zwei Modi „Modelling“ und „Simulation“ prototypisch implementiert und in einem Simulationsszenario evaluiert.

### ABSTRACT

This article describes an approach in which the modeling and simulation process for the indoor climate of buildings takes place entirely in a virtual reality (VR) environment. In this way, the simulation expert is immersively involved with regard to the spatial design, spatial perception and interaction with the spatial model. The simulated indoor climate resulting from the interactions and other variables such as Energy requirements are visualized in the VR environment as a feedback to the user.

To implement the described approach in terms of software, the Modelica tool Dymola was integrated with the game engine Unity to form a complete system, with bidirectional data exchange taking place between the two programs. The building model of the Modelica library BuildingSystems forms the basis of the building physics calculations in Dymola. In Unity, a head-mounted display was integrated to implement the VR simulation environment, and the two modes "modeling" and "simulation" were prototypically implemented and evaluated in a simulation scenario.

### EINLEITUNG

Werkzeuge zur energetischen Gebäude- und thermischen Raumsimulation ermöglichen detaillierte Prognosen des Innenraumklimas und des Gebäudeenergiebedarfs. Je nach Simulationstool oder Workflow stehen dem Anwender hierfür unterschiedliche Werkzeuge zur Modellbildung (Definition von Gebäudegeometrie und -topologie, der Baukonstruktion sowie der Abbildung des Nutzerverhaltens) und zur Durchführung und Auswertung der Simulationsanalyse zur Verfügung.

Bei diesem Vorgehen findet sowohl die Erzeugung als auch die thermisch-energetische Analyse des Gebäudemodells innerhalb eines Workflows statt, bei dem der Simulationsexperte nicht immersiv bzgl. seiner Aktionen und seiner Wahrnehmung in die Simulationsumgebung integriert ist. Der Nutzer hat eine körperliche Distanz zum Modellierungsprozess und zum Simulationsexperiment, in dem er dem Simulationswerkzeug über ein Keyboard seine Aktionen mitteilt und über den Bildschirm den Fortschritt des Modellierungsprozesses und die Ergebnisse des Simulationsergebnisses wahrnimmt.

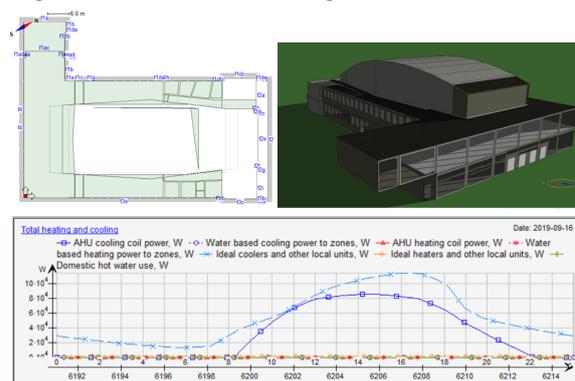


Abbildung 1: Klassische Modellierung und energetische Simulationsanalyse von Gebäuden mit IDA ICE als Desktop-Anwendung

Abbildung 1 zeigt am Beispiel des Konzertsaals der UdK Berlin diese heute übliche Vorgehensweise beim Modellierungs- und Simulationsprozess am Beispiel des Simulationswerkzeuges IDA ICE (IDA ICE, 2020). Der Simulationsexperte modelliert zunächst mit der Desktop-Anwendung die Geometrie des Gebäudes unter Berücksichtigung seiner Zonierung (oder importiert es alternativ als BIM-Modell) und

legt danach die bauphysikalischen Eigenschaften der Baukonstruktion sowie die Raumnutzungen fest. Im nächsten Schritt erfolgt für den gegebenen Klimastandort die Simulationsanalyse, woraufhin die Ergebnisse auf dem Bildschirm als Zeitreihen oder visualisiert im 3D-Gebäudemodell dargestellt werden.

Der in diesem Beitrag vorgestellte methodische Ansatz will diese Arbeitsweise grundlegend verändern, in dem er den Simulationsexperten sowohl beim Modellierungsprozess als auch bei der Durchführung des Simulationsexperimentes immersiv in eine VR-Simulationsumgebung einbettet. Der komplette Workflow ausgehend von der Formgebung des einzelnen Raumes über die Materialdefinition der Baukonstruktion bis hin zur Durchführung und Auswertung des Simulationsexperiments findet nun im virtuellen Raum statt, in welchem dem Nutzer die hierfür notwendigen Werkzeuge zur Verfügung gestellt werden müssen.

Für den ersten Teil des beschriebenen Ansatzes wurden bereits erste Softwaretools entwickelt, welche eine direkte oder indirekte Modellierung von Architekturmodellen in VR-Umgebungen unterstützen.



Abbildung 2: Direktes Erstellen von Architekturmodellen in der Virtual Reality mit VR Sketch

Abbildung 2 zeigt das kommerzielle Sketchup-Plugin VR Sketch (VR Sketch, 2020), welches dem Nutzer ein Toolset an Modellierungswerkzeugen anbietet, das eine *direkte Modellierung* von 3D-Gebäudemodellen in einer VR-Umgebung ermöglicht. Schon während des Entstehungsprozesses kann das 3D-Modell virtuell begangen werden, wodurch eine interaktive Modellentwicklung unterstützt wird.

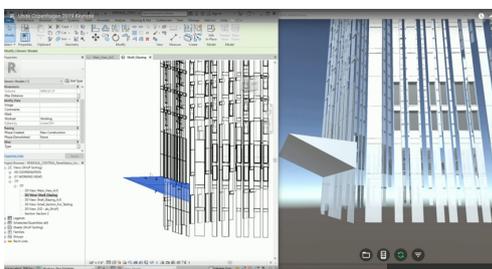


Abbildung 3: Indirektes Modellieren in einer VR-Umgebung in Revit über Unity Reflect

Das Werkzeug Unity Reflect (Unity, 2019) ist ein neues, ebenfalls kommerzielles Plugin für die Spiele-Engine Unity (Unity, 2020), welches den Realtime-Datenaustausch zwischen im Bauwesen genutzten

CAD-Programmen (bisher für Revit und SketchUp) und Unity unterstützt und damit ein *indirektes Modellieren in einer VR-Umgebung* erlaubt.

Wie in Abbildung 3 dargestellt, ermöglicht dies zukünftig ein kollaboratives Arbeiten in Entwurfsteams: ein Architekt nimmt geometrische Anpassungen an seinem BIM-Modell im CAD-Werkzeug vor (z.B. im Revit-Modell in Abbildung 3 links), welche unmittelbar in der auf Unity basierenden VR-Simulationsumgebung aktualisiert werden (siehe Abbildung 3 rechts). Ein Kollege kann dann die Modellveränderung in der virtuellen Realität im realen Maßstab unmittelbar erfahren und bewerten und daraufhin dem entwerfenden Architekten am Desktop-Rechner sein Feedback geben.

Beide VR-Umgebungen unterstützen bisher keine thermische und energetische Analyse von Innenräumen. Sie erlauben zudem dem Nutzer auch keine Interaktion mit dem Gebäudemodell, er ist also bisher ein passiver Betrachter von selbst geschaffenen architektonischen Räumen, deren Echtzeitvisualisierungen auf Basis der verwendeten modernen Hardware- und Softwaretechnologien aus der Gaming-Industrie eine hohe Qualität aufweisen.

Ein erster Forschungsansatz zum Computational Steering im Bereich der Innenraumklimasimulation demonstrierte die Realzeitkopplung eines parallelisierten echtzeitfähigen CFD-Rechenkerns mit einer interaktiven VR-Visualisierungsumgebung (Treec et al., 2006).

Der Prototyp einer ersten Realzeitkopplung zwischen einem thermischen Mehrzonen-Gebäudemodell auf Basis von Modelica/Dymola und einer VR-Simulationsumgebung auf Basis von Unity3D/HTC Vive wird in (Nytsch-Geusen et al., 2017b) beschrieben. Bei diesem Ansatz wurden zunächst das in einem CAD-Tool extern erstellte BIM-Modell nach Unity importiert, dort manuell um Möglichkeiten zur Nutzerinteraktion angereichert und danach durch manuelle Konfiguration bzgl. des Datenaustausches mit dem energetischen Gebäudemodell verbunden.

### ANFORDERUNGEN AN DIE VR-SIMULATIONSUMGEBUNG

Ausgehend von dem beschriebenen Entwicklungsstand wurde von den Autoren zunächst die Anforderungen an eine VR-Umgebung zur thermischen Raumsimulation spezifiziert, um diese anschließend in ihren wesentlichen Grundzügen prototypisch zu implementieren. Die VR-Simulationsumgebung soll zwei voneinander getrennte Modi unterstützen, den *Modelling-Mode* und den *Simulation-Mode*.

#### **Modelling-Mode**

Im Modelling-Mode sollen die *Raumgeometrie* und die *bauphysikalischen Eigenschaften der Baukonstruktion* definiert werden können. Hierzu müssen dem Nutzer Werkzeuge zur Verfügung

gestellt werden, die eine einfache Definition von architektonischen Räumen innerhalb einer VR-Umgebung unterstützen (vgl. Abbildung 4).

Weiterhin soll die *Auswahl des lokalen Standorts* des zu analysierenden Gebäudes erfolgen können, was zum einen die Festlegung des Außenklimas bedeutet, prinzipiell aber auch die Berücksichtigung der Gebäudeumgebung und des Gebäudeuntergrunds (Nachbarbebauung und Topographie des Standorts) mit einschließen muss, um die Randbedingungen des Simulationsexperiments (z.B. Verschattung im urbanen Raum) realistisch abbilden zu können.



Abbildung 4: Definition der Raumgeometrie im Modelling-Mode innerhalb der prototypisch implementierten VR-Simulationsumgebung

Ein weitere notwendige Funktionalität des Modelling-Modes ist die *Festlegung der möglichen Interaktionen* im Raummodell während des Simulationsexperiments. Hierzu gehört das Ermöglichen des Öffnens von Türen und Fenstern, die Bereitstellung von Schaltern und Kontrollfunktionen für die Beleuchtung, die Klimatisierung von Räumen (Raumthermostat) und das Aktivieren des Sonnenschutzes von Fassaden.

### Simulation-Mode

Nach der Definition des Gebäudemodells im Modelling-Mode soll in einfacher Weise in den Simulation-Mode gewechselt werden können, um in einer Realzeit-Simulation das sich einstellende Innenraumklima zu analysieren. Da thermische Prozesse wie die Erwärmung oder Abkühlung eines Raumes oder die Veränderung der täglichen Sonnenposition nur sehr langsam stattfinden, muss der Simulation-Mode auch eine zeitweise erhebliche *Beschleunigung der Realzeitsimulation* über einen Realzeitfaktor unterstützen. Auf diese Weise kann der Nutzer entweder in einer beschleunigten Simulation die physikalisch korrekt berechneten thermischen Prozesse passiv beobachten oder er kann in einer Realzeitsimulation aktiv das Simulationsmodell in seinen wesentlichen Parametern beeinflussen und so die Auswirkungen auf das Raumklima analysieren.

Hierzu soll der Simulationsexperte im Modelling-Mode mit den reaktiven Elementen des Raummodells interagieren können, wie z.B. durch das vollständige Öffnen oder Kippen eines Fensters oder das Aktivieren eines Sonnenschutzes vor einem Fenster.

Außerdem soll es möglich sein, während des Simulationsexperiments auch die *Raumgeometrie* und somit die Dimensionen und Ausrichtung der opaken und transparenten Baukonstruktionen *dynamisch anzupassen*. Hierdurch soll eine direkte interaktive Analyse des Innenraumklimas und der zugehörigen Energieeffizienz für geometrisch unterschiedliche Raumausprägungen ermöglicht werden.

Ein weitere Anforderung an den Simulations-Mode besteht in der *Visualisierung der berechneten raumklimatischen sowie energetischen Zustände des Raummodells*. Je nach Fragestellung des Simulationsexperiments und Detaillierungsgrad des genutzten Raummodells sollen die Raumlufttemperatur und Raumluftfeuchte, die Strahlungstemperaturen der einzelnen Bauteiloberflächen, die Außenlufttemperatur und Außenluftfeuchte des Standorts, aber auch energetische Größen wie die notwendige Heiz- oder Kühlleistung zur Aufrechterhaltung eines gewünschten Raumklimas als räumlich verteilte Momentanwerte oder Zeitreihen über den gesamte Simulationszeitraum dargestellt werden können.

### Softwarearchitektur

Die Softwarearchitektur der interaktiven VR-Simulationsumgebung ist nach einem streng modularen Ansatz aufgebaut (vgl. Abbildung 5).

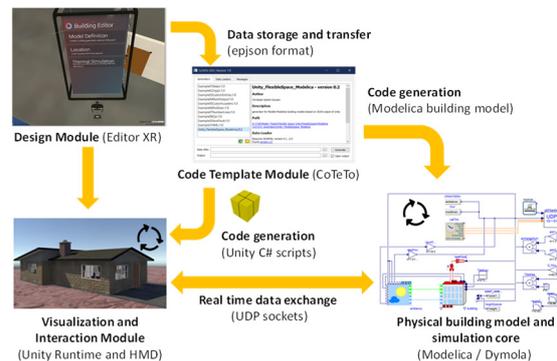


Abbildung 5: Softwarearchitektur der interaktiven VR-Simulationsumgebung

Das **Design Module** enthält die Modellierungstools, mit welchen im Modelling-Mode das 3D-Gebäudemodell in der VR-Umgebung erstellt werden kann, was die Festlegung seiner Geometrie und Topologie, seiner Baukonstruktionen, den lokalen Standort und möglicher Modell-Interaktionen einschließt.

Beim Wechsel vom Modelling- in den Simulation-Mode werden über einen Datelexport (**Data storage and transfer**) alle notwendigen Daten zur automatisierten Erzeugung des thermischen Gebäudemodells an das **Code Template Module** weitergeleitet. Dieses generiert zum einen das über das Design Module definierte thermische Modelica-Gebäudemodell (**Physical bulding model**), zum anderen die hierzu passenden Skriptdateien, welche Umfang und Struktur des bidirektionalen Datenaustausches zwischen dem thermischen Gebäudemodell und dem **Visualization and Interaction Module** zur Laufzeit konfigurieren.

Diese Modul ist ebenfalls eine VR-Umgebung, die alle Anforderungen des Simulation-Mode, wie Nutzerinteraktion sowie Anpassung des Realzeitfaktors und der Raumgeometrie unterstützt.

### IMPLEMENTIERUNG

Die Implementierung der VR-Simulationsumgebung beruht zum einen auf der Game-Engine Unity (Unity, 2020). Unity verfügt über einen Szenen-Editor, in dem komplexe 3D-Szenen über Szenengraphen definiert werden können und über einen Play-Mode, in welchem ein Nutzer auf Grundlage der vorab definierten Szenen sich in 3D-Räumen bewegen und mit den Objekten der Szene (GameObjects) interagieren kann. Die in Unity enthaltenen Mechanismen können über C#-Skripte erweitert werden, um die Logik innerhalb der Szenen im Play-Mode zu erweitern. Weiterhin unterstützt Unity die Integration von VR/AR-Geräten wie z.B. des Head Mounted Display HTC Vive oder die Augmented Reality Brille Microsoft HoloLens.

Die strukturelle Aufteilung von Unity in den Szenen-Editor und den Play-Mode wurde genutzt, um den Modelling- und Simulation-Mode in Unity umzusetzen. Durch Nutzung von Unity Ergänzungsmodulen und C#-Skriptprogrammierung wurde der Szenen-Editor zum Modelling-Mode erweitert und in den Play-Mode die gewünschten Funktionen des Modelling-Modus implementiert.

Die Modellierung des physikalischen Gebäude-modells inkl. der bidirektionalen Datenkommunikation zu Unity über das UDP-Datenprotokoll erfolgt über die Modelica-Bibliotheken BuildingSystems (Nytsch-Geusen et al., 2016) und Modelica\_DeviceDrivers (Thiele et al., 2017) in Kombination mit dem Simulationswerkzeug Dymola. Das Modelica-Gebäudemodell wird in Abhängigkeit der im Design Module vorgenommenen Modellierung automatisch über das CoTeTo-Werkzeug zur Codegenerierung (Nytsch et al., 2017a) erzeugt.

### Design Module

Die Implementierung des Design Module wurde auf Grundlage der Unity-Erweiterung Editor XR (Editor XR, 2020) vorgenommen, welche den Editier-Mode von Unity so erweitert, das alle Modellierungsfunktionen von Unity (z.B. Erstellen von Szenen, Instanzieren und Manipulation von Szenenobjekten bzgl. ihrer Größe, Position, Form und visueller Darstellung) innerhalb einer VR-Umgebung durchgeführt werden können. Weiterhin bietet die Editor-XR-Toolbox vordefinierte Werkzeuge, wie z.B. 3D-Menüs oder eine 3D-Workbench, welche an die Anforderungen der Implementierung eines Unity-Projekts flexibel angepasst werden können.

Auf dieser Grundlage wurden über C#-Skripte das Nutzer-Interface und die Modellierungswerkzeuge zur Gebäudeerstellung für das Design Module implementiert (vgl. Abbildungen 6 und 7).

Abbildung 6 oben links zeigt das Nutzerinterface mit den drei Hauptmenüpunkten

- Modelldefinition (Model Definition),
- Wahl des Simulationsstandorts (Location) und
- Wechsel in den Simulation-Mode (Thermal Simulation)



Abbildung 6: Nutzerinterface des Design Module

Die anderen gezeigten Menübeispiele ermöglichen die Wahl eines geografischen Standorts, die Auswahl eines Werkzeugs zur Raummodellierung (Wände, Fenster, Türen) bzw. die individuelle Festlegung der thermischen Kapazitäten der einzelnen opaken Gebäudeelemente über die Bauteildicke.

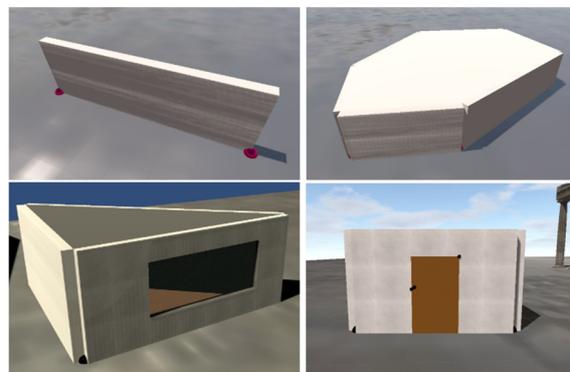


Abbildung 7: Erstellen von Raumgeometrien und Gebäudeelementen im Design Module

Nach Auswahl des Wand-Editierwerkzeugs kann mit Hilfe der Controller des genutzten Head-Mounted-Displays HTC Vive über die Markierung eines Polygonzuges auf dem in der virtuellen Welt sichtbaren Boden die Länge und Ausrichtung der einzelnen Wände eines Raum definiert werden, wobei die Wände hierbei eine unterschiedliche Dicke aufweisen können. Die Bodenplatte und die Decke werden nach Schliessen des Polygonzuges automatisch ergänzt (vgl. Abbildung 4 sowie Abbildung 7 oben links und rechts).

Anschließend können in die einzelnen Wände Fenster und Türen eingefügt werden, die jede für sich ein für den Nutzer interaktives Gebäudeelement im Simulation-Mode darstellen können (Tür öffnen bzw. Fenster öffnen durch Drehen oder Kippen).

### Datentransfer von Unity nach CoTeTo

Nach Definition eines Gebäudemodells im Design Module werden die das Gebäudemodell beschreibenden Geometrie-, Material-, und Standortdaten im JSON-Format gespeichert und exportiert. Hierbei wird auf das spezielle epJSON-Format des international weit verbreiteten Gebäudesimulationsprogramm EnergyPlus (Adams, 2018) aufgebaut, welches seit der Version 8.8 experimentell und seit der Version 9.0 standardmäßig als Eingabeformat dient.

Die Verwendung dieses „Quasistandards“ hat den Vorteil, dass auch ein für EnergyPlus erstelltes Gebäudemodell in den Workflow der VR-Simulationsumgebung nach CoTeTo importiert und zur Erzeugung eines Modelica-Gebäudemodells genutzt werden kann. Beim Datenexport aus dem Design Module wurde das epJSON-Format in seiner Struktur um einen Abschnitt erweitert, welcher ergänzende topologische Informationen zu den Bauelementen des Einzelraumes beschreibt.

### Code Template Module

Im Design Module wird durch Betätigen des Menü-Eintrages „Generate Model“ das erweiterte JSON-File im EnergyPlus-Format erzeugt, von CoTeTo eingelesen und daraufhin das Modelica-Gebäudemodell generiert. Dieses kann über die Menüeinträge „Compile Model“ und „Run Model“ zunächst in ein ausführbares Modelica-Modell übersetzt und anschließend das Simulationsexperiment gestartet werden (vgl. siehe Abbildung 8).

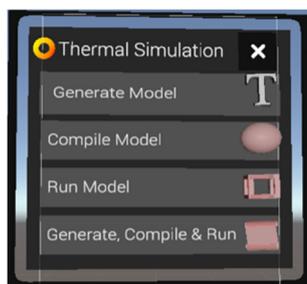


Abbildung 8: Menü zum Generieren, Kompilieren und Starten des Modelica-Gebäudemodells

Alternativ können auch alle drei Schritte über den Menüeintrag „Generate, Compile & Run“ ausgeführt werden, wobei die VR-Umgebung dann automatisch in den Simulation-Mode wechselt.

Softwaretechnisch geschieht dies durch das Aufrufen eines Python-Skripts, welche alle Einzelschritte des Workflows über entsprechende Python-Schnittstellen von CoTeTo und Dymola ansteuert (vgl. Codeausschnitt in Abbildung 9). Das Python-CoTeTo-Modul erzeugt dabei nicht nur über ein Modelica-Template das Gebäudemodell, sondern

auch über C#-Templates zwei C#-Skripte. Mit diesen Skripten wird in Abhängigkeit vom Simulationsszenario festgelegt, welche Variablenwerte vom Modelica-Modell an die 3D-Szene im Simulation-Mode in Unity gesendet werden (Datum, Uhrzeit, Längen- und Breitengrad sowie Wetterdaten des gewählten Standorts, Oberflächen- und Raumlufttemperaturen des Gebäudemodells, Umgebungslufttemperaturen etc.) und welche Werte aus der 3D-Unity-Szene zurück an das Modelica-Modell übermittelt werden sollen (Nutzerinteraktionen wie das Öffnen von Türen, und Fenstern oder das Verändern der Raumgeometrie etc.).

```
# Model generation based on CoTeTo
from CoTeTo.Controller import Controller
con = Controller()
gen = con.generators['Unity_FlexibleSpace_Modelica::0.2']
gen.execute(,buildingData.epJSON', './FlexibleSpace')

# Real-time simulation experiment with Dymola
from dymola.dymola_interface import DymolaInterface
dym = DymolaInterface()
dym.openModel('BuildingSystems/package.mo')
dym.openModel('FlexibleSpace/package.mo')
dym.openModel('Modelica_DeviceDrivers/package.mo')
dym.translateModel('FlexibleSpace')
dym.simulateModel(problem='FlexibleSpace', ...)
```

Abbildung 9: Python-Skript zur Steuerung des Workflows (Erzeugung und Übersetzung des Modelica-Modells, Starten der Realzeit-Simulation)

### Physical building model

Für die Anforderung einer flexiblen Raumgeometrie zur Laufzeit wurden das Zonenmodell sowie die Wand-, Tür- und Fenstermodelle der BuildingSystems-Bibliothek um die optionale Möglichkeit einer flexiblen Bauteilgeometrie und Orientierung erweitert. Abbildung 10 demonstriert wie in Abhängigkeit des Werts des Flags `geometryType` das Zonen- und Wandmodell eine starre oder flexible Geometrie aufweisen kann.

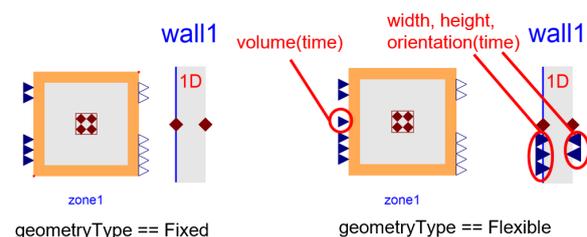


Abbildung 10: Zonen- und Wand-Modell der Modelica BuildingSystems-Bibliothek für starre (links) und flexible (rechts) Raumgeometrien

Ist das Flag auf den Wert `Flexible` gesetzt, können die zusätzlich zur Verfügung gestellten Modellinputs für das Raumluftvolumen des Zonenmodells sowie die Höhe, Breite, der Azimut- und Neigungswinkel der Wand-, Tür- und Fenstermodelle mit anderen dynamischen Werten belegt werden. Diese können z.B. über den Datenaustausch aus dem parallel



Durchführung der Simulationsexperimente der Realzeitfaktor um ein vielfaches vergrößert wurde. Alle Bauteiltemperaturen und die Raumlufttemperatur sind mit einer Temperatur von 20 °C initialisiert.

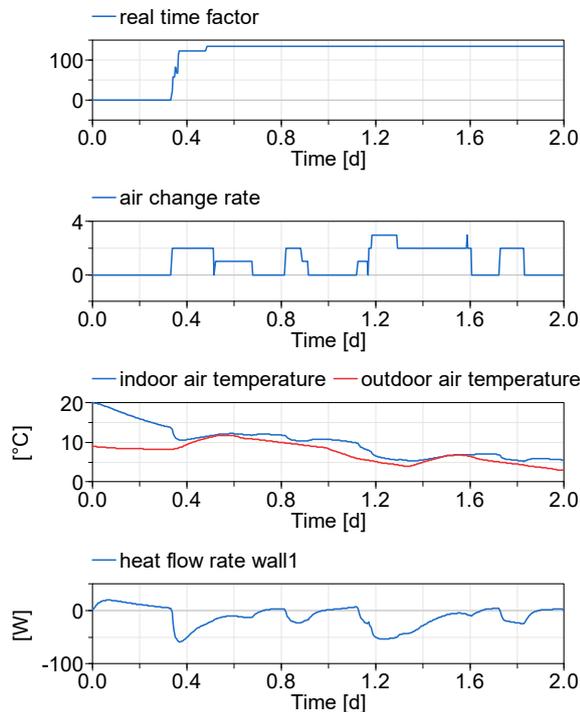


Abbildung 15: Einfluss des Nutzerverhaltens auf das Innenraumklima

Der in Abbildung 15 dargestellte erste Anwendungsfall zeigt deutlich den Einfluss des Nutzerverhaltens durch das Öffnen der Tür bzw. Öffnen oder Kippen des Fensters. Immer wenn ein erhöhter Luftwechsel durch das Nutzerverhalten verursacht wird, nähert sich die Raumlufttemperatur der kalten winterlichen Außenlufttemperatur stärker an und der Wärmestrom von der noch gegenüber der Raumluft wärmeren Baukonstruktion in den Innenraum erhöht sich zeitweise (exemplarisch für eine Wand dargestellt). Die Luftwechselsimulation ist in diesem Experiment noch physikalisch vereinfacht abgebildet: Eine geöffnete Tür führt im Modell zu einem Luftwechsel von 1/h, ebenfalls ein gekipptes Fenster, ein über den Drehflügel geöffnetes Fenster zu einem Luftwechsel von 2/h. Falls die Tür und das Fenster gleichzeitig geöffnet sind, werden beide Luftwechsel addiert.

Im zweiten Anwendungsfall wird ausgehend vom gleichen Raummodell der Grundriss während des Simulationsexperiments vom Nutzer dreimal verändert, wobei sich das Raumluftvolumen zunächst zweimal deutlich vergrößert und danach wieder etwas verringert wird (exemplarisch ist auch von der Wand wall12 die Fläche dargestellt, die sich wegen der Grundrissänderung ebenfalls ändert, vgl. Abbildungen 16 und 17)<sup>17</sup>. Der Wert der Raumlufttemperatur beträgt entsprechend der Sollwert-Vorgabe konstant 20 °C, entsprechend zeigt die benötigte Heizlast einen

<sup>17</sup> Sowohl der Wert des Raumluftvolumens als auch der Wandfläche von wall2 weisen im Fall der flexiblen Geometrie für einen

sprunghaften Anstieg bzw. eine Reduzierung, wenn vom Nutzer im Simulation-Mode der Grundriss des Raummodells vergrößert bzw. verkleinert wird.

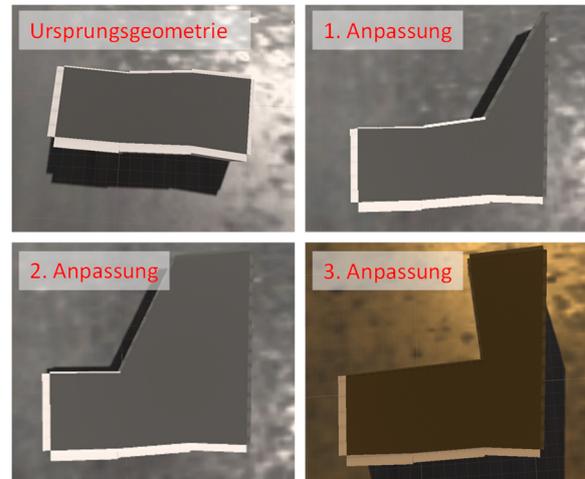


Abbildung 16: Raummodell mit flexibler Raumgeometrie: ursprünglicher Grundriss und seine Modifikationen während des Simulationsexperiments

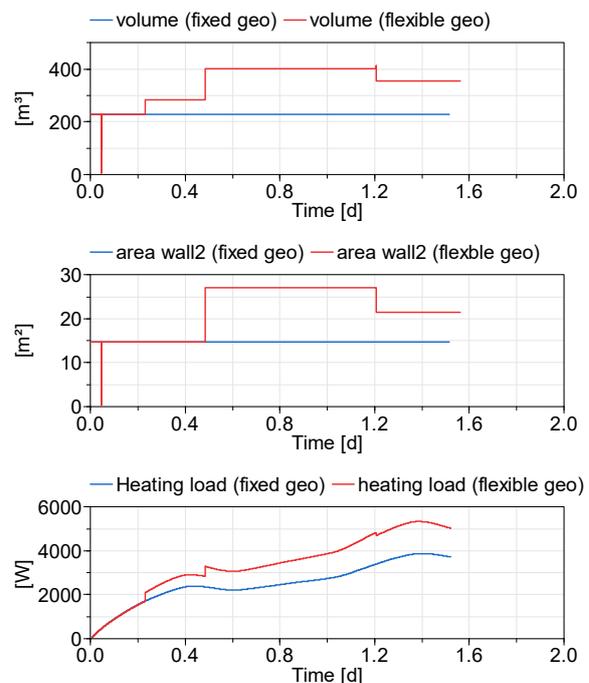


Abbildung 17: Einfluss einer flexiblen Raumgeometrie auf die Heizlast

### ZUSAMMENFASSUNG

Es konnte an Hand eines Softwareprototypen grundsätzlich gezeigt werden, dass der vollständige Workflow vom Modellieren bis zum Simulieren eines thermischen Raummodells in einer VR-Simulationsumgebung stattfinden kann, wodurch der Simulationsexperte immersiv in den Workflow einbezogen wird. Hierzu gehören zum einen die Realisierung des Modelling-Modes mit an die Virtual Reality angepassten Modellierungswerkzeugen, zum

Moment einen sehr niedrigen Wert auf, was an einer kurzen Störung beim Datenaustausch zwischen Unity und Dymola liegt.

anderen die zur Bereitstellung eines Simulation-Modes mit den Möglichkeiten zur Realzeit-Simulation inkl. eines anpassbaren Realzeitfaktors, der Nutzerinteraktion mit reaktiven Gebäudeelementen sowie zur Anpassung der Raumgeometrie während des Simulationsexperiments.

### AUSBLICK

Ausgehend vom vorliegenden Softwareprototypen der VR-Simulationsumgebung sollen im nächsten Schritt Erweiterungen hinzugefügt werden, die zum einen die Modellierungsmöglichkeiten deutlich erweitern, aber auch optional den Detaillierungsgrad des physikalische Gebäudemodells verfeinern lassen. Hierzu gehören zunächst die Erweiterung des Modelling-Modes um die Definitionsmöglichkeit von Einzonnen- auf Mehrzonengebäudemodelle sowie eine verfeinerte Auswahlmöglichkeit der thermischen Eigenschaften der baukonstruktiven Elemente, was bereits in der genutzten Modelica-BuildingSystems-Bibliothek möglich ist. In einem weiteren Schritt soll ein räumlich diskretisiertes Modelica-Raummodell (Mucha, 2017) integriert werden, um mit der VR-Umgebung auch räumlich differenzierte Aussagen zum Innenraumklima treffen zu können.

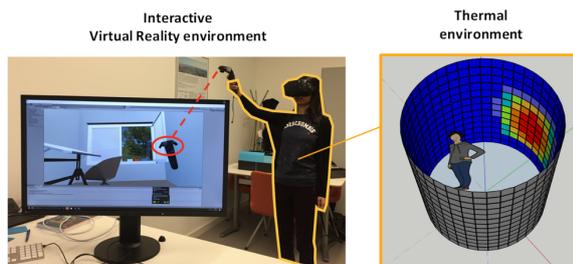


Abbildung 18: Thermisches Display als Ergänzung für die interaktive VR-Simulationsumgebung

Im laufenden Forschungsprojekt „GEnEff - Neuartige Bewertung der Gebäude-Energie-Effizienz und innovative Demonstration mittels Simulationsmethoden und Virtual Reality“, soll in den nächsten Jahren ein thermisches Display entwickelt werden, mit welchem der Nutzer die im bauphysikalischen Raummodell berechneten Luft- und Strahlungstemperaturen direkt körperlich wahrnehmen kann (vgl. Abbildung 18). Hierdurch soll der immersive Eindruck bei der Analyse des bauphysikalischen Verhaltens von Innenräumen in der VR-Umgebung noch erheblich gesteigert werden.

### DANKSAGUNG

Ein Teil der in diesem Beitrag dargestellten Forschungsergebnisse wurden im Projekt „EnEff BIM: Planung, Auslegung und Betriebsoptimierung von energieeffizienten Neu- und Bestandsbauten durch Modellierung und Simulation auf Basis von Bauwerkinformationsmodellen“ (Förderkennzeichen: 03ET1177D, gefördert vom Bundesministerium für Wirtschaft und Energie) erarbeitet.

### LITERATUR

- Adams, M. B., New, J. R. 2018. EnergyPlus Performance Improvements via JSON Input Refactoring. Proceedings Building Performance Analysis Conference and SimBuild 2018, Chicago, USA.
- Thiele, B., Beutlich, T., Waurich, V., Sjölund, M., Bellmann, T. 2017. Towards a Standard-Conform, Platform-Generic and Feature-Rich Modelica Device Drivers Library. Proceedings 12th International Modelica Conference, Prag, Tschechien.
- Editor XR 2020: Unity-Erweiterung zur Unterstützung von Modellierung von Unity-Szenen in VR oder AR. GitHub-Repository <https://github.com/Unity-Technologies/EditorXR>
- IDA ICE 2020. IDA Indoor Climate and Energy simulation tool. Webseite: <https://www.equa.se/en/ida-ice>, letzter Zugriff 9.3.2020.
- Mucha, K. 2017. Ein Simulationsansatz zur Bewertung von Hitzestressrisiken in Innenräumen – Weiterentwicklung eines zonalen Modells in Modelica. Dissertation, Fakultät Gestaltung, Universität der Künste Berlin.
- Nytsch-Geusen, C., Banhardt, C., Inderfurth, A., Mucha, K., Möckel, J., Rädler, J., Thorade, Tugores, C. 2016. BuildingSystems – Eine modular hierarchische Modell-Bibliothek zur energetischen Gebäude- und Anlagensimulation. Proceedings BAUSIM 2016 IBPSA. Dresden, Deutschland.
- Nytsch-Geusen, C., Inderfurth, A., Kaul, W., Mucha, K., Rädler, J., Thorade, M., Tugores, C. 2017a. Template based code generation of Modelica building energy simulation models. Proceedings 12th International Modelica Conference, Prag, Tschechien.
- Nytsch-Geusen, C., Möckel, J., Rädler, J., Thorade, M. 2017b. BuildingSystems\_VR - A new approach for immersive and interactive building energy simulation. Proceedings Building Simulation Conference 2017, San Francisco, USA.
- Unity 2020. Echtzeit-3D-Entwicklungsplattform Unity. Webseite: <https://unity.com/de>, letzter Zugriff 9.3.2020.
- Unite 2019. Ausschnitt aus der Keynote der Unite 2019 Conference, Kopenhagen, Dänemark.
- Treeck, C., Wenisch, P., Borrmann, A., Pfaffinger, M., Egger, M., Wenisch, O., Rank, E. 2006. Towards interactive indoor thermal comfort simulation. Proceedings ECCOMAS CFD 2006, TU Delft.
- VR 2020. Erweiterung für Sketchup zur Modellierung in der Virtual Reality. Webseite VRsketch <https://vrsketch.eu/>, letzter Zugriff 9.3.2020