

BBSR-
Online-Publikation
16/2022

Crowd Safety

von

Ivan Bratoev
Jakob Fellner
Nick Förster
Prof. Dr.-Ing. Frank Petzold
Dr.-Ing. Gerhard Schubert



Crowd Safety

Interaktive Entwurfsplanung unter Berücksichtigung von
Personenströmen und deren Sicherheitsaspekten

Gefördert durch:



Bundesministerium
für Wohnen, Stadtentwicklung
und Bauwesen

aufgrund eines Beschlusses
des Deutschen Bundestages

ZUKUNFT BAU
FORSCHUNGSFÖRDERUNG

Dieses Projekt wurde gefördert vom Bundesinstitut für Bau-, Stadt- und Raumforschung (BBSR) im Auftrag des Bundesministeriums für Wohnen, Stadtentwicklung und Bauwesen (BMWSB) aus Mitteln des Innovationsprogramms Zukunft Bau.

Aktenzeichen: 10.08.18.7-18.12

Projektlaufzeit: 04.2019 bis 09.2021

IMPRESSUM

Herausgeber

Bundesinstitut für Bau-, Stadt- und Raumforschung (BBSR)
im Bundesamt für Bauwesen und Raumordnung (BBR)
Deichmanns Aue 31–37
53179 Bonn

Fachbetreuer

Bundesinstitut für Bau-, Stadt- und Raumforschung
Referat WB 3 „Forschung im Bauwesen“
Dr.-Ing. Arnd Rose
arnd.rose@bbr.bund.de

Autoren

Technische Universität München, Lehrstuhl für Architekturinformatik

Ivan Bratov, M. Sc.
ivan.bratoev@tum.de

Jakob Fellner, M. Sc.
jakob.fellner@tum.de

Nick Förster, M. A.
nick.foerster@tum.de

Prof. Dr.-Ing. Frank Petzold
petzold@tum.de

Dr.-Ing. Gerhard Schubert
schubert@tum.de

Stand

Februar 2021

Bildnachweis

Titelbild: Technische Universität München, Lehrstuhl für Architekturinformatik
Weitere Bildnachweise: siehe Seite 73

Vervielfältigung

Alle Rechte vorbehalten

Der Herausgeber übernimmt keine Gewähr für die Richtigkeit, die Genauigkeit und Vollständigkeit der Angaben sowie für die Beachtung privater Rechte Dritter. Die geäußerten Ansichten und Meinungen müssen nicht mit denen des Herausgebers übereinstimmen.

Zitierweise

Petzold, Frank; Schubert, Gerhard; Bratov, Ivan; Fellner, Jakob; Förster, Nick; Crowd Safety: Interaktive Entwurfsplanung unter Berücksichtigung von Personenströmen und deren Sicherheitsaspekten.
BBSR-Online-Publikation 16/2022, Bonn, Mai 2022.

Inhaltsverzeichnis

Abstract	5
1 Einführung	6
2 Forschungsansatz	9
2.1 Eingabe	10
2.2 Simulation	11
2.3 Visualisierung	12
3 Lösungsansatz	13
3.1 CDP // Collaborative Design Platform	13
3.2 crowd:it – Personenstromsimulationen	15
4 Anwendungskonzept	18
4.1 Anwendungsszenarien	18
4.2 Eingabe	23
4.3 Simulation	28
4.4 Visualisierung	33
5 Technische Umsetzung	47
5.1 Technische Grundlagen	48
5.2 Informationsaustausch	49
5.3 Eingabe	52
5.4 Simulation	54
5.5 Visualisierung	55
6 Evaluierung	57
6.1 Konventioneller Prozess	57
6.2 Automatisierter Prozess	63
6.3 Diskussion	65
6.4 Quantitativer Vergleich	65
6.5 Qualitativer Vergleich	66
7 Zusammenfassung und Ausblick	68
A1 Abkürzungsverzeichnis	70
A2 Abbildungen	71
A3 Tabellenverzeichnis	72
A4 Quellen- und Literaturverzeichnis	73

Abstract

The project Crowd Safety – Including Pedestrian Flows and Security Aspects in Planning scrutinizes the integration of agent-based simulations in the early design stages of architecture and urban design. On the one hand, it builds upon the microscopic pedestrian model crowd:it. On the other, the project uses the CDP // Collaborative Design Platform, which is dedicated explicitly to contingent design negotiations. The presented research couples these elements through a server-client architecture, integrating computation-intensive simulations as agile design support.

Based on this foundation, we examine the requirements for pedestrian simulations in early contingent and dynamic design phases and investigate user interactions, simulation, and visualization concepts. We discuss how to control pedestrian simulation through the intuitive and tangible interactions of the CDP. Additionally, we describe the adaptation of the simulation process to real-time interactions and ad-hoc results. Lastly, we discuss visualization methods to respond directly to design input with intuitive feedback.

This report documents these findings and the prototypical implementation of these concepts. It describes the implementation of the server-client architecture and the communication protocol as a plug-in to the CDP. Furthermore, the conversion of user interactions to simulation input and the visualization of results are discussed in more detail. Based on this prototype, we conduct both a technical and design-centred evaluation of the presented concepts. The project validates the possibility to integrate high-accuracy pedestrian simulations in the early design phases. Thus, it reveals the potentials of crowd simulations as agile decision support.

The project's success can be attributed to the help provided to us by our partner, accu:rate GmbH.

1 Einführung

Fußgängerbewegungen und Nutzungsauslastung sind entscheidende Faktoren für die Leistungsfähigkeit, Sicherheit und Qualität von öffentlichen Räumen und Gebäuden. Belebte Plätze und hochfrequentierte Innenräume müssen auch bei hohen Personendichten zügigen Einlass ermöglichen, Komfort gewährleisten und in Notsituationen eine rasche Räumung erlauben. Personenbewegungen korrekt einzuschätzen, ist somit ein kritisches Planungskriterium für den Neubau und den Betrieb öffentlicher Bauwerke. Besonders grundlegende Entwurfsentscheidungen, wie die Wahl des Erschließungssystems oder die Nutzungsverteilung wirken sich stark auf Personenbewegungen und die spätere Auslastung von Gebäuden aus. Diese frühen Entwurfsentscheidungen sind frühe Weichenstellungen, welche die Planung maßgeblich beeinflussen.

Gleichzeitig ist die Entscheidungsgrundlage in frühen Planungsphasen fragmentarisch und unsicher. Entwurfsentscheidungen wirken sich auf zahlreiche, heterogene Planungskriterien aus. Als „Wicked Problems“, haben Entwurfsprobleme Horst Rittel (Rittel und Webber 1973) zufolge keine einzelne und eindeutige Lösung. Vielmehr erfordern sie die Verhandlung verschiedener Planungsbeteiligter – Expert*innen, Nutzer*innen und tangierte Stakeholder. Unter Berücksichtigung relevanter Informationen werden Lösungsalternativen diskutiert und iteriert, bis die relevanten Entwurfsziele zufriedenstellend erfüllt sind. Insbesondere in Bezug auf komplexe und uneindeutige Faktoren – wie Personenbewegungen – mangelt es in frühen Entwurfsphasen an belastbaren Informationen. Lediglich statische Normen (z.B. Notausgangsverordnungen, Mindestbreiten) und die individuelle Entwurfserfahrung stehen als Entscheidungsgrundlage zur Verfügung. Keine der beiden Optionen sichert jedoch eine evidenzbasierte, kontextuell optimale Lösung.

Computersimulationen können dieser Unsicherheit entgegenwirken, indem sie Personenströme und Nutzungsmuster prognostizieren. So machen Simulationen planungsrelevante Informationen bereits während dem Entwurfsprozess als Entscheidungsgrundlage verfügbar. Planer*innen können Simulationsergebnisse in den Entwurfsprozess miteinbeziehen und gegen andere relevante Kriterien abwägen. Stresstests werden vor dem ersten Spatenstich möglich, Betriebskonzepte und Prozesse können aktiv entworfen werden. Je früher relevante Informationen im Entwurfsprozess zur Verfügung stehen, desto signifikanter können sie den weiteren Planungsprozess und letztendlich die Qualität der baulichen Lösung beeinflussen (Roetzel 2015). Sind Simulationsergebnisse bereits in den frühen Entwurfsphasen verfügbar, so beeinflussen sie grundlegende Entwurfsentscheidungen und verhelfen zu einem optimierten Entwurfsergebnis. Andererseits steigt mit dem Fortschreiten des Planungs- und Bauprozesses zunehmend der Aufwand, die bestehende Planung anzupassen (Serginson et al. 2013). Entwerferische Fehlentscheidungen können im Nachgang nur mit sehr aufwändigen Maßnahmen kompensiert werden – durch die Anpassung zahlreicher Folgeentscheidungen oder kostspielige Kompensationsmaßnahmen, wie zusätzliche Fluchtwege und umständliche Sicherheitskonzepte.

Allerdings stehen die umfassenden Inputparameter für akkurate Simulationen in frühen Entwurfsphasen oft noch nicht zur Verfügung oder sind vage und unvollständig. Darüber hinaus sind Simulationstools meist nicht an kontingente und dynamische Entwurfsprozesse angepasst (Bratoev et al. 2018; Ritter und Schubert 2014): Nutzereingaben sind kontraintuitiv und zeitintensiv. Der Berechnungsvorgang ist unflexibel und nicht interaktiv. Simulationsergebnisse werden nicht intuitiv verständlich im Entwurfskontext präsentiert. Daher werden derzeitige Personenstromsimulationen vorwiegend zur Validierung (oder Falsifizierung) finalisierter Planungen eingesetzt. Ihr Potenzial, grundlegende Entwurfsentscheidungen zu unterstützen, wird noch nicht ausgeschöpft.

Der Entwurf und die Nutzungssimulationen von Gebäuden und öffentlichen Räumen sind aktuell voneinander losgelöste iterative Prozesse. Erst wenn eine Planung so weit vorangeschritten ist, dass sie den Ansprüchen der Simulationssoftware genügt, kann sie in Hinblick auf Personenströme evaluiert werden. Wenn die Simulation Probleme aufzeigt, muss die Planung adaptiert – oder im schlechtesten Fall – verworfen werden. Dies bedeutet einen wenig agilen Optimierungsprozess in langen, ineffizienten Iterationen aus Anpassung und Evaluierung (Petersen und Svendsen 2010). Um dieser Diskrepanz entgegenzuwirken, müssen Personenstromsimulationen bereits während frühen Entwurfsphasen verfügbar sein. Das vorgestellte Forschungsprojekt schlägt eine Brücke zwischen frühen Entwurfsentscheidungen und ihren Auswirkungen auf die spätere Gebäudenutzung. Hierfür muss die Simulation nahtlos in den Entwurfsprozess eingebettet werden. Aus diesem Grund koppelt das Projekt akkurate Personenstromsimulationen mit Entwurfswerkzeugen, die in kollaborativen Planungsprozessen zum Einsatz kommen. So wird eine evidenzbasierte Entscheidungsunterstützung unmittelbar mit kreativen Arbeitsmethoden verbunden.

Projektbeschreibung

Das Projekt baut einerseits auf der *CDP // Collaborative Design Platform* auf, die in den letzten zehn Jahren an der Technischen Universität München entwickelt wurde (Schubert 2014). Andererseits basiert es auf der Simulationssoftware *crowd:it*, die von dem KMU *accu:rate*, einem Spin-Off der TU München entwickelt wurde (accu:rate 2021). Um diese Software als agiles Werkzeug in Planungsprozesse zu integrieren, wurde eine Schnittstelle entwickelt, mithilfe derer Personenstromsimulationen in die *CDP* eingebunden werden (Abbildung 1).

Im Rahmen des Projektes wurden Konzepte erarbeitet, um dieses Simulationstool an die Anforderungen früher, dynamischer Entwurfsphasen anzupassen. Verschiedene Nutzungsszenarien dienten als Grundlage für die Konzeptfindung. Eingabe-, Simulations- und Visualisierungsmethoden für die Entscheidungsunterstützung in kollaborativen Entwurfsszenarien wurden untersucht: Geometrische und haptische Interaktionen wurden als Simulationsinput erforscht. Konzepte zur Adaption der Simulation an dynamische Entwurfsprozesse wurden entwickelt. Zudem wurde eruiert, wie Simulationsergebnisse in Echtzeit als Planungshinweise ausgegeben werden können.

Zur Umsetzung dieses Ziels wurden sowohl Anpassungen auf der Hardware wie auf der Softwareseite unternommen. Spezifische Interaktionsmethoden zur Steuerung der Simulation wurden zur *CDP* hinzugefügt.

Die Plattform wurde adaptiert, um Innenräume adäquat abzubilden. Darüber hinaus wurde eine Open Source-Kommunikationsschnittstelle erarbeitet, welche die *CDP* mit einem performanten Simulationsserver verbindet. Somit wird sichergestellt, dass trotz flexibler Anforderungen an das Endgerät die notwendige Rechnerleistung für aufwändige Simulationen zur Verfügung steht. Zwischen den beiden Systemen können somit schnell und verlustfrei große Datenmengen ausgetauscht werden. Darüber hinaus wurden Visualisierungsstrategien erforscht, die Simulationsergebnisse qualitativ verständlich und responsiv wiedergeben. Im Rahmen von Nutzungsstudien wurden die erarbeiteten Lösungsansätze qualitativ und quantitativ evaluiert.

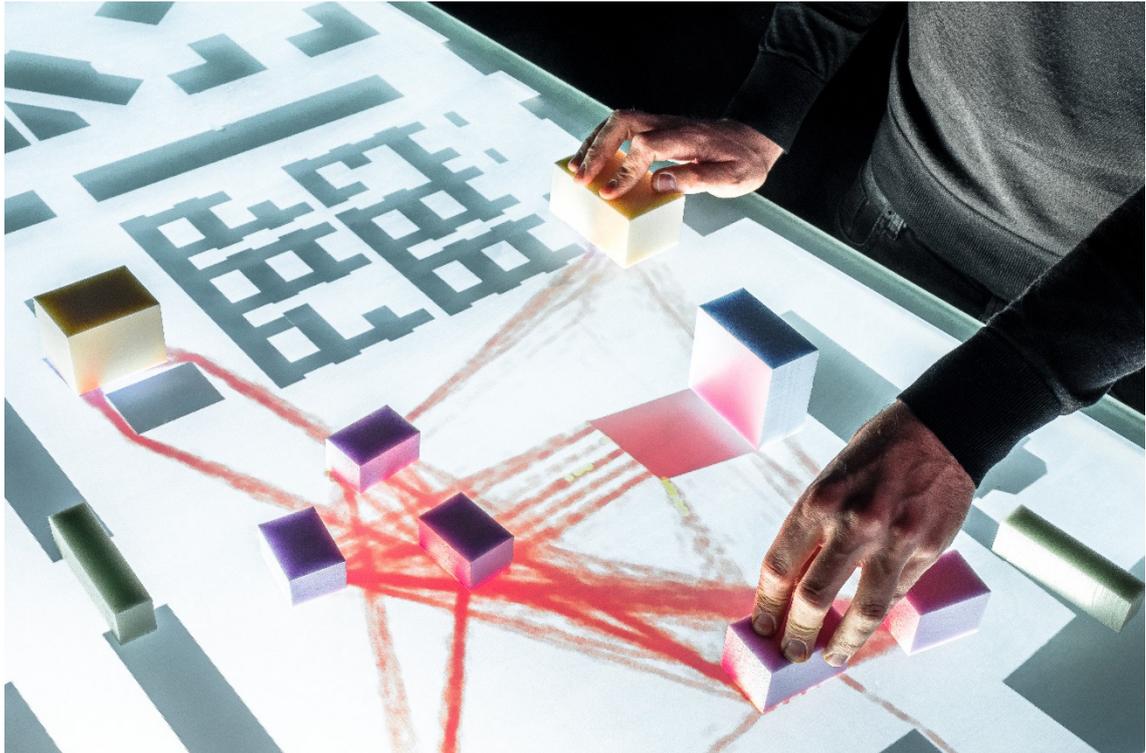


Abbildung 1 Prototypische Umsetzung (Quelle: Lehrstuhl für Architekturinformatik)

2 Forschungsansatz

Eine nahtlose Integration von entscheidungsunterstützenden Simulationen in den Entwurfsprozess verspricht signifikante Vorteile: einerseits Zeit- und Kostenoptimierung während der Planung, andererseits eine optimierte Performance entwerfener Gebäude und öffentlicher Räume. Planungsfehler werden vorweggenommen, und somit aufwändige Änderungsmaßnahmen minimiert. Andererseits wird die Flächeneffizienz, Aufenthaltsqualität und Sicherheit von entworfenen Räumen gesteigert, da Informationen zu Personenbewegungen als Entscheidungskriterium zur Verfügung stehen. Darüber hinaus unterstützt die Einbettung evidenzbasierter Methoden in kollaborativen Entwurfsprozessen eine transparente und konsensuelle Entscheidungsfindung. Simulationsergebnisse unterstützen die beteiligten Akteure in ihrer Argumentation, ohne sie zu bevormunden. Um diese Ziele zu erreichen müssen *Eingabe*, *Simulation* und *Visualisierung* adaptiert werden. Insofern geht das Projekt von folgenden Hypothesen aus:

Eingabe:

Durch eine Anpassung der Mensch-Computer-Interaktion können Personenstromsimulationen als entwurfsunterstützende Werkzeuge in frühe Planungsphasen integriert werden. Die Simulation wird nahtlos in die Entwurfsinteraktionen eingebunden.

Simulation:

Die Auslagerung der Simulation auf einen externen *Server* sichert Flexibilität in der Anwendung. Einerseits werden geringere Anforderungen an die Leistung des Eingabegeräts gestellt, andererseits ist das Zuschalten hoher Rechenkapazität für aufwändige Simulationen möglich.

Visualisierung:

Mithilfe geeigneter Visualisierungsstrategien können auf der Grundlage von Zwischenergebnissen Tendenzen angezeigt und in Echtzeit als entwurfsunterstützendes, qualitatives Feedback zurückgegeben werden.

Aufbauend auf diesen Hypothesen wurden im Rahmen dieses Projekts drei Forschungsschwerpunkte definiert: *Eingabe*, *Simulation* und *Visualisierung*. Im Kontext der *Eingabe* wurden geeignete Eingabemethoden und -parameter zur Steuerung der Simulation untersucht. Dieses Thema ist eng mit dem zweiten Schwerpunkt *Simulation* verbunden. Es wird untersucht, wie die Simulation durch die Kommunikationsschnittstelle adaptiert werden kann, um den Anforderungen früher Entwurfsphasen zu entsprechen. Im Zusammenspiel von *Eingabe* und *Simulation* zählt, eine Balance zwischen der Einfachheit und Intuitivität des Inputs und möglichst akkuraten Simulationsergebnissen zu finden. Der Schwerpunkt *Visualisierung* umfasst Methoden zur Präsentation von Simulationsergebnissen im Entwurfskontext sowie Strategien zur Reaktion auf Interaktionen in Echtzeit.

2.1 Eingabe

Die notwendigen Eingabeparameter für akkurate Personenstromsimulationen sind in frühen Entwurfsphasen sehr unsicher und nur fragmentarisch vorhanden. Insofern mussten Wege gefunden werden, um Simulationen mit unvollständigen Eingabedaten zu steuern. Darüber hinaus erfordert die Nutzereingabe klassischer Simulationstools technisches Vorwissen und ist somit nicht für Entwurfsteilnehmer*innen unterschiedlicher fachlicher Hintergründe geeignet. Daher müssen intuitive Eingabemethoden entwickelt werden, die sich nahtlos mit Entwurfsinteraktionen verbinden. Die Eingabe von Simulationsparametern muss unaufwändig sein – die darf nicht vom Entwurfsprozess ablenken oder diesen verzögern. Um dynamische, kollaborative Prozesse zu unterstützen, müssen Nutzerinteraktionen – anders als bei einer Desktopanwendung – hierarchielos und simultan möglich sein.

Insofern wird erforscht, wie Simulationsmodelle intuitiv und automatisiert aufgebaut werden können. Darüber hinaus wird untersucht, wie Inputparameter vereinfacht (gemittelt) oder vorweggenommen werden können. Eingaben in der *CDP* müssen als Simulationsparameter interpretiert und an den Simulationsserver kommuniziert werden. Im Bereich *Eingabe* ergeben sich somit folgende Forschungsfragen:

1. Welche Inputparameter müssen für eine vereinfachte Bedienung definiert werden?
2. In welcher Form kann/muss die Eingabe notwendiger Parameter erfolgen, um die Unterbrechungen des Entwurfsprozesses zu minimieren?
3. Für welche Parameter können Annahmen und Vereinheitlichungen getroffen werden, ohne das Simulationsergebnis signifikant zu verfälschen?

2.2 Simulation

Um störungsfreie Interaktionen zu gewährleisten, muss der Simulationsvorgang adaptiert werden. Statt ein gänzlich neues Verfahren für Personenstromsimulationen zu entwickeln, wurde eine Echtzeit-Rückkopplung von Simulationsergebnissen anhand der bestehenden Software *crowd:it* untersucht. Diese wurde, wie beschrieben, mittels eines Kommunikationsprotokolls mit der *CDP* verbunden. Mithilfe dieser Schnittstelle werden Eingabeparameter übergeben, Berechnungen gestoppt und gestartet, sowie Simulationsergebnisse an die *CDP* zurückgegeben. Anpassungen an dem Simulationsvorgehen wurden im Rahmen dieser Schnittstelle erforscht.

Um frühe Entwurfsphasen zu unterstützen muss die Simulation an den Entwurfsprozess angepasst werden. Die Simulation muss in Echtzeit unvorhersehbare Entwurfsinteraktionen reagieren. Die Simulation darf den Entwurfsprozess nicht durch passive Berechnungszeiten behindern. Im Gegenteil muss sie durchgehend Interaktionen mit dem Simulationsmodell zulassen und auf diese umgehend mit Feedback reagieren.

Insofern ergeben sich im Rahmen der *Simulation* folgende Forschungsfragen:

1. Wie können Echtzeit-Interaktionen bei gleichzeitig möglichst korrekten Berechnungsergebnissen erreicht werden?
2. Wie kann die Kommunikationsschnittstelle mit der Simulation verknüpft werden, um kontinuierliche Interaktionen (auch während laufenden Berechnungen) zu ermöglichen?
3. Wie werden große Mengen von Informationen (Eingaben und Ergebnisse) über ein Kommunikationsprotokoll möglichst schnell/kompakt ausgetauscht?
4. Welche Berechnungen erfolgen durch das lokale Eingabegerät, welche durch den Simulationsserver?

2.3 Visualisierung

Neben den Eingabeparametern bildet die Ausgabe (Darstellung der Ergebnisse) den zweiten direkten Schnittpunkt zur Nutzer*in. Auf der *CDP* müssen Simulationsergebnisse adäquat als Entscheidungsgrundlage präsentiert werden. Hierfür wurden Darstellungsmethoden erforscht, die dem *Level of Detail* früher Entwurfssimulationen gerecht werden und für verschiedene Stakeholder im Entwurfskontext verständlich sind. Vorversuche zeigten, dass finale Simulationsergebnisse nicht ad-hoc verfügbar sind. Um Entwurfsinteraktionen mit responsivem Feedback zu beantworten, mussten Methoden untersucht werden, um vorläufige und unvollständige Simulationsergebnisse darzustellen.

Insofern lauten die zentralen Fragestellungen im Bereich Visualisierung:

1. Welche Simulationsergebnisse sind in frühen, kollaborativen Planungsphasen relevant?
2. Wie und mit welchem Detailgrad können qualitative Ergebnisse als Entscheidungsgrundlage präsentiert werden?
3. Wie können unmittelbar nach einer Interaktion vorläufige Ergebnisse dargestellt werden? Wie können diese Ergebnisse mit zunehmendem Rechenfortschritt an ein finales Resultat angenähert werden?

3 Lösungsansatz

Die Integration von Personenstromsimulationen in frühe Entwurfsprozesse wurde auf Grundlage zweier bestehender Projekte untersucht. Diese bieten eine Lösung für jeweils einen Teil der Problemstellung: Die *CDP // Collaborative Design Platform* (Schubert 2014) ermöglicht die Kombination klassischer analoger und neuer digitaler Entwurfsmethoden in kollaborativen Situationen. Über das beschriebene Kommunikationsprotokoll wird die Personenstromsimulationssoftware *crowd:it* (accu:rate 2021) in die Plattform eingebunden und somit für den Entwurfsprozess erschlossen. Die Verbindung der beiden Tools bildet die Grundlage für das vorgestellte Projekt.

3.1 CDP // Collaborative Design Platform

Der sinnvolle Einsatz des Rechners bereits in den frühen Phasen der architektonischen Planung (z.B. städtebaulicher Maßstab 1:500) ist die zentrale Problemstellung der Forschungsgruppe *CDP // Collaborative Design Platform* (Abbildung 2). Seit über 10 Jahren untersucht diese Gruppe am Lehrstuhl für Architekturinformatik der Technischen Universität München Mensch-Maschine-Interaktionen im Entwurfskontext (Schubert 2014). Die *CPD* diente als Basis für Forschungsprojekte zur Kopplung physischer und digitaler Elemente (Schubert 2014), Augmented Reality (Schubert et al. 2016), Virtual Reality (Schubert et al. 2019) und Plug-In Frameworks (Petzold und Auer 2018) in kollaborativen Planungskontexten. Die Integration verschiedener Simulationstools in frühe Entwurfsphasen wurde unter anderem anhand von Solarpotenzialanalysen (Bratoev et al. 2018) und Gebäudesimulationen (Ritter und Schubert 2014) untersucht. Insofern bietet die Plattform bereits ein Inventar an Interaktionsformen, die als Input für Simulationstools geeignet sind und die Anforderungen früher Entwurfsphasen erfüllen.



Abbildung 2 CDP // Collaborative Design Platform – etablierte Entwurfswerkzeuge und digitale Tools nahtlos verbunden (Quelle: Lehrstuhl für Architekturinformatik)

Ausgehend von den Anforderungen des Entwurfs-, Planungs- und Kommunikationsprozesses wurde die CDP im Rahmen der Dissertation *Interaktionsformen für das digitale Entwerfen* (Schubert 2014) als *proof of concept* entwickelt. Auf Basis der Analyse des Entwurfsprozesses und den damit verbundenen Anforderungen an Entwurfswerkzeuge wurde eine Design-Plattform konzipiert und prototypisch umgesetzt. Im Kern steht die direkte Kopplung von real greifbaren Volumenmodellen und Handskizzen mit interaktiven digitalen Inhalten wie Analysen und Simulationen. Durch die Verwendung etablierter Entwurfswerkzeuge als Eingabemedien (*Tangible User Interface, TUI*) wird neben der Eingabe vager Informationen durch Handskizze und Arbeitsmodell vor allem der Prozess des „Visual Thinkings“ und somit der Ideenfindung ermöglicht. Das wird auch benutzt, um die Schnittstelle zu integrieren)

Grundsätzlich besteht die CDP aus einem berührungssensitiven Tisch-Display. Physische Objekte werden in Echtzeit von einer Tiefenkamera mit zusätzlicher RGB-Kamera gescannt und als digitale Modelle rekonstruiert. So dienen die Objekte als Input-Geometrien für Entwurfstools und Berechnungen. Gleichzeitig wird das physische Modell auf dem horizontalen Bildschirm mit digitalen Informationen erweitert. Der Tisch verfügt über ein Plug-In Framework, über das verschiedene Tools in die Plattform eingebunden werden. Dieses wird auch genutzt, um mittels einer Kommunikationsschnittstelle Personenstromsimulationen in die CDP einzubinden.

Auf Basis der Analyse des Entwurfsprozesses und den damit verbundenen Anforderungen an Entwurfswerkzeuge wurde eine Design-Plattform konzipiert und prototypisch umgesetzt – im Kern steht die direkte Kopplung von real greifbaren Volumenmodellen und Handskizzen mit interaktiven digitalen Inhalten wie Analysen und Simulationen.

Im Vergleich zu herkömmlichen *Tangible User Interfaces* dienen die physischen Arbeitsmodelle nicht nur als Steuerungselemente, sondern auch als Abbild von Gebäuden. Durch diese direkte Verbindung von physischer

und digitaler Welt, sind die realen Objekte als Gesamtkörper mit der Simulation verbunden und werden zu unmittelbaren Akteuren im digitalen Entwurfsszenario (Schubert 2014). Plastische Verformungen der physischen Modelle, wie etwa das Zerschneiden oder Verformen, aber auch Positionsänderungen werden direkt in der Szene aktualisiert und in den Echtzeitberechnungen berücksichtigt. Hierdurch erweitert sich der momentan vorhandene Informationsgehalt rein physischer Arbeitsmodelle und Skizzen um zusätzliche digitale Informationsebenen. Dadurch ist es möglich, in gewohnter Art und Weise Arbeitsmodelle zu erstellen und mittels dieser zu entwerfen. Die zusätzliche Echtzeitinteraktion ermöglicht es der Entwerfer*in, direkte Folgen des Entwurfes berechnen und darstellen zu lassen.

3.2 crowd:it – Personenstromsimulationen

Personenstromsimulationen prognostizieren das Verhalten und die Bewegung von Fußgänger*innen auf der Grundlage mathematischer Modelle (Abbildung 3). Somit können ausgehend von Input-Parametern und einer Input-Geometrie Voraussagen zur Befüllung und Auslastung von Verkehrs- und Aufenthaltsflächen getroffen werden. Sie haben somit einen entscheidenden Vorteil gegenüber Entwurfsverfahren, die auf Norm- oder Erfahrungswerten basieren: Sie ermöglichen es, situative Problemstellen einer Planung herauszuarbeiten und durch gezielte Maßnahmen zu entschärfen.



Abbildung 3 *crowd:it* – Simulationsmodell (Quelle: *accu:rate*)

Simulationsmodell

Abhängig von Maßstab und Anwendungsbereich, die eine Fußgängersimulation adressiert, kommen verschiedene Berechnungsmodelle zum Einsatz. In der Dissertation *Methoden zur Abbildung menschlichen Navigationsverhaltens* bei der Modellierung von Fußgängerströmen unterscheidet Angelika Kneidl zwischen unterschiedlichen Verfahren zur Abbildung menschlichen Bewegungs- und Navigationsverhaltens (Kneidl 2013): Simulationen können einerseits als **makroskopische** und **mikroskopische** Modelle kategorisiert werden, andererseits können wir zwischen räumlich **diskreten** und **indiskreten** Modellen differenzieren.

Makroskopische Modelle vernachlässigen lokale Geschehnisse zugunsten großmaßstäblicher Muster und Kenngrößen. Klassische Beispiele für makroskopische Simulationsmodelle sind **Netzwerkflussmodelle**, **Kontinuumsmodelle** und **strömungsbasierte Modelle** (Kneidl 2013). Diese Modelle nähern Gruppenbewegungen durch abstrakte Gesetze, wie Potenzialfelder und oder analog zur Fluidodynamik an. Diese Berechnungsansätze entfalten ihre Stärke vor allem bei hohen Personendichten und in der großmaßstäblichen Betrachtung (Kneidl 2013).

Mikroskopische Simulationsmodelle beziehen psychologische und soziologische Aspekte von Personenbewegungen mit in die Berechnung ein. Hierzu werden Fußgänger*innen einzeln modelliert. So können individuelles Bewegungsverhalten und Interaktionen zwischen Fußgänger*innen abgebildet werden. Um dieses Ziel zu erreichen, bauen diese Simulationen auf *Partikelsystemen*, *Zellulären Automaten* und *Multi-Agenten-Systemen* auf (Dietrich et al. 2014). **Zelluläre Automaten** stellen Fußgängerbewegungen als Zustände von Zellen in einem diskreten Raster dar (Kretz 2007). Das **Social Force Model** bildet individuelles Fußgängerverhalten analog physikalischer Anziehung und Abstoßung nach (Bitzer 2010). Das **Optimal Steps Model** von Seitz und Köster (Seitz und Köster 2012; Kneidl 2013) kombiniert diese Methoden, indem physikalische Anziehung und Abstoßung als Navigationsfelder interpretiert werden, welche die indiskrete Bewegung von „Fußgänger“-Partikeln durch den Simulationsraum steuern. Im Gegensatz zu räumlich **diskreten** Modellen, wie dem zellulären Automaten, bewegen sich diese Partikel in einem kontinuierlichen **indiskreten** Simulationsraum. Insofern können Bewegungsgeschwindigkeit, Richtung und Größe der simulierten Personen deutlich präziser moduliert werden. Agenten können unterschiedliche Schrittgrößen und Eigenschaften zugewiesen werden (Seitz 2016). Somit lassen sich verschiedene Personengruppen mit unterschiedlichen Eigenschaften modellieren.

Der Kern der Software *crowd:it* wurde im Rahmen der Dissertation von Angelika Kneidl entwickelt (Kneidl 2013). Um eine möglichst realistische Darstellung von Personen- und Gruppenbewegungen zu erreichen, wurde untersucht, wie sich Personen fortbewegen und mit ihrer Umwelt interagieren. Erkenntnisse aus der Sozialpsychologie und Simulationstechniken wurden aufeinander abgestimmt (Kneidl 2013).

Das resultierende Modell greift auf das *Optimal Steps Model* (Seitz und Köster 2012; Sivers et al. 2014; Sivers und Köster 2015) als Grundlage für ein *Multi-Agenten-System* zurück (accu:rate 2021). Somit werden Fußgänger*innen als autonom agierende Agenten modelliert, die komplexe Verhaltensregeln befolgen, individuelle Eigenschaften haben, sich gegenseitig beeinflussen und spezifisch definierte Ziele anstreben (Kneidl 2013). Dieser Ansatz wurde um dynamische Flutungsfelder erweitert, um das Navigationsverhalten von Fußgänger*innen akkurat abzubilden (Kneidl 2013).

Anwendungsszenario

Ein klassisches Anwendungsszenario der Simulation *crowd:it* ist die Evakuierung eines Bauwerks über ein oder mehrere Geschosse oder einer öffentlichen Fläche (accu:rate 2021). In diesem Fall simuliert die Software, wie eine maximal zulässige Anzahl an Personen, ein Gebäude oder eine Fläche verlässt. Die Simulation zeigt, ob alle Personen den Gefahrenbereich in der vorgeschriebenen Zeit verlassen können sowie ob es zu Staus und problematischen Verdichtungen kommt. Weitere Anwendungen von Fußgängersimulationen sind Effizienz- und Komfortanalysen (accu:rate 2021). Mit deren Hilfe kann beispielsweise überprüft werden, wie schnell ein Raum mit Personen gefüllt werden kann. Die Simulation prognostiziert Bottlenecks, welche die Performance und Aufenthaltsqualität eines Bauwerks beeinträchtigen. Zudem evaluieren Komfortanalysen eine Einrichtung im Betrieb. Hier werden nicht nur Stauungen deutlich, sondern auch die Auslastung von Nutzungsbereichen, wie Verkaufsständen oder Toiletten. So werden Kapazitätsgrenzen erkennbar.

Die Simulation *crowd:it* wird aktuell als Standalone-Desktopanwendung genutzt. Um eine Planungsvariante zu simulieren, wird diese zunächst als DXF-Zeichnung eingelesen. Diese Datei enthält sowohl geometrische wie semantische Informationen: Durch die Layerstruktur werden Elemente als Simulationsobjekte – Quellen, Zwischenziele oder Ziele – klassifiziert. Wenn im Zeichenprogramm Entwurfsänderungen vorgenommen werden, muss die DXF-Zeichnung neu eingelesen werden. In diesem Fall erkennt *crowd:it* adaptierte Geometrien anhand ihrer ID und übernimmt festgelegte Einstellungen. Im Anschluss wird ausgehend von jeder Quelle die Zielreihenfolge der Agenten manuell definiert (Quelle => Zwischenziel(e) => Ziel). Darüber hinaus können verschiedene Einstellungen der Agenten (Größe, Geschwindigkeit) und des Simulationsmodells (Dynamische Navigation, Auflösung des Simulationsraums, etc.) definiert werden. Die Simulation kann gestartet werden, sobald alle Inputparameter festgelegt wurden. Die Berechnung läuft nach Runden ab, in denen jeder Agent jeweils einen Schritt weiterbewegt wird. Während dieser Berechnung lässt das Programm keine Interaktionen zu. Geometrische Adaptionen oder die Veränderung von Eingabeparametern erfordern den Abbruch und Neustart der Simulation. Sobald sämtliche Simulationsschritte vollständig durchlaufen sind, werden die Simulationsergebnisse dargestellt. Hierfür stehen die Animation der Agenten, die Wegpfade der Agenten, sowie verschiedene *Heatmaps* und Statistiken zur Verfügung.

4 Anwendungskonzept

Um Personenstromsimulationen mit kollaborativen Entwurfswerkzeugen zu verknüpfen, wurden die vorgestellten Projekte – *CDP* und *crowd:it* – verknüpft und aufeinander angepasst. Auf Basis der beschriebenen Forschungsfragen wurden neue Interaktionsverfahren, Simulationsabläufe und Visualisierungsmethoden für Personenstromsimulationen erforscht. Zunächst wurden zwei Anwendungsszenarien für Personenstromsimulationen in den frühen Entwurfsphasen definiert. Diese Szenarien stellen spezifische Anforderungen an die Forschungsfelder *Eingabe*, *Simulation* und *Visualisierung*. Ausgehend von diesen Anforderungen wurden Lösungskonzepte entwickelt, welche auf die Problemstellungen des Nutzungsszenarios reagieren.

4.1 Anwendungsszenarien

Zum besseren Verständnis der Anwendungssituation und konkreter Anforderungen wurden zwei Nutzungsszenarien definiert. Diese Szenarien wurden so gewählt, dass sie ein möglichst breites Spektrum an Anwendungsfällen und Kriterien abdecken, in Bezug auf Entwurfsproblem, Anwenderkonstellationen, Maßstäblichkeit, Zeiteffizienz, Entwurfsinteraktionen und Planungskriterien. Das erste Szenario – **Innenraum** – betrachtet die Adaption eines Bestandsgebäudes anhand einer hochfrequentierten Erdgeschosszone. Das zweite Szenario – **Außenraum** – beinhaltet die temporäre Umplanung eines zentralen öffentlichen Raumes.

Beide Szenarien verkörpern ein vielversprechendes Anwendungsgebiet für die Nutzung von Personenstromprozessen in kollaborativen Entwurfsdiskussionen. Einerseits sind Personenbewegung und Auslastung in beiden Szenarien wesentlich für die Sicherheit und Performance des entworfenen Raums. Andererseits umfassen beide Anwendungsszenarien eine Vielzahl an Planungsbeteiligten, die über eine Vielzahl miteinander verbundener Planungsparameter entscheiden müssen. Insofern ist es wichtig, Personenstromsimulationen in diesem Entscheidungs- und Abstimmungsprozesse verfügbar zu machen. Die Planungsbeteiligten erstellen gemeinsam Varianten und können Simulationsergebnisse umgehend in ihre Argumentation einbinden. Drängende Fragen, wie die Kollision von Nutzungsmustern oder Sicherheitsbedenken können gemeinsam abgewogen und geklärt werden. In beiden Anwendungen sind Sicherheitsanalysen (z.B. sichere Evakuierung im Brandfall) sowie Komfortanalysen (z.B. effizienter und staufreier Einlass und Betrieb) relevant.

Szenario 1: Innenraum

Als praxisrelevantes Beispiel für interaktive Personenstromsimulationen im Innenraum eignet sich eine Nutzungsänderung am Gebäudebestand (Abbildung 4).

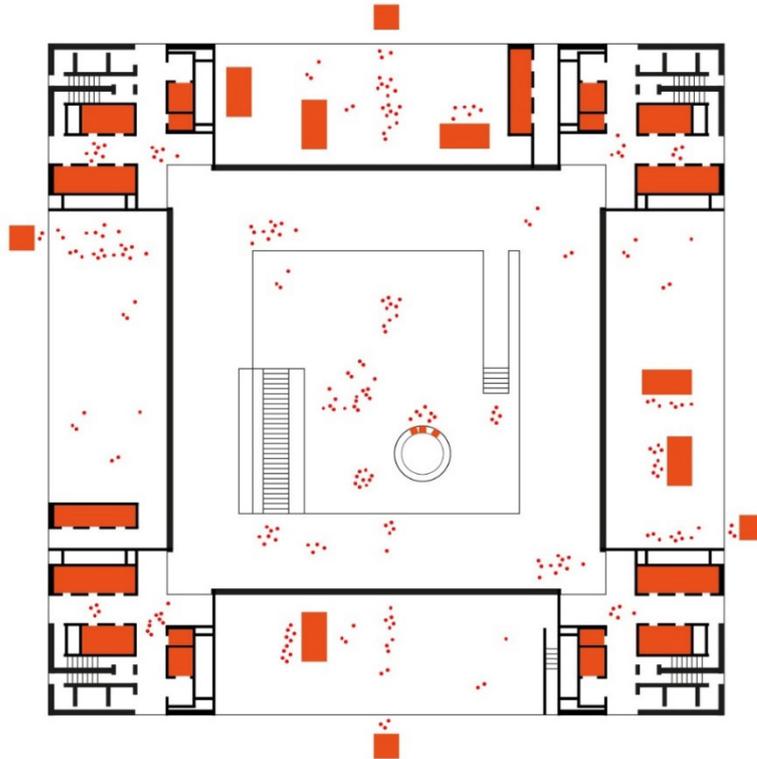


Abbildung 4 Szenario „Indoor“ – Hochhaus (Quelle: Lehrstuhl für Architekturinformatik)

Da Planungsentscheidungen die Gebäudestruktur betreffen, haben sie einen sehr langfristigen Einfluss auf den Gebäudebetrieb. Die Integration von Personenstromsimulationen in den Entwurfsprozess kann somit kostspielige Änderungen in der Nutzung vorwegnehmen. Die Simulation muss somit Grundlagen für langfristige Entscheidungen verlässlich kommunizieren.

Für die Adaption bestehender Gebäude analysieren Planungsbeteiligte – Architekt*innen, Fachplaner*innen, Bauherr*innen – Personenströme innerhalb eines existierenden Gebäudeentwurfes. Die Rahmenbedingungen für Entwurfsinteraktionen sind stark durch den Gebäudebestand eingeschränkt – die Geometrie von Ein- und Ausgängen, die Tragstruktur und bestehende Bauauflagen sind limitierende Rahmenbedingungen für Entwurfsinteraktionen. Insofern konzentriert sich der Entwurf auf gezielte Eingriffe und die Umnutzung von Gebäudebereichen.

Da geometrische Änderungen durch eine Vielzahl von Abhängigkeiten beschränkt sind, konzentriert sich die Entwurfsinteraktion auf die räumliche Programmierung, die Definition von Raumfunktionen oder das (de)aktivieren von Ausgängen und Treppenhäusern. Durch die Verringerung von Ausgängen kann getestet werden, ob die Entfluchtung im Evakuierungsfall gegeben ist. Ein weiterer Anwendungsfall ist ein Funktionstest des Gebäudes während Wartungsarbeiten mit vorübergehend abgesperrten Bereichen.

Die Interaktion fokussiert somit auf funktionale Anpassungen. Personenstromsimulationen haben das Potenzial diese hochkomplexen und kontingente Entscheidungsfindung zu unterstützen: Sie können Informationen zu schwer vorhersehbaren Interferenzen von Nutzungen aufzeigen, die Kapazitätsgrenzen bestimmter Gebäudeteile eruieren.

Interaktionen: Die Gebäudegeometrie, Ein- und Ausgänge sowie aktuelle Nutzungen (Toiletten, Infopoints) sind im Plangebiet vordefiniert – sie werden durch die *CDP* geladen. Entwerfende interagieren vorwiegend, indem sie Ein- und Ausgänge sowie Nutzungen hinzufügen oder (de-)aktivieren. Darüber hinaus führen Planer*innen kleinere Anpassungen an der Gebäudestruktur durch: Wände werden hinzugefügt, Durchbrüche geplant.

Elemente: Im Innenraumszenario bilden die Ein- und Ausgänge eines Gebäudes, sowie die vertikale Erschließung (Treppen, Rolltreppen, Aufzüge) die Zugänge, über die Fußgänger den Simulationsbereich betreten. Nach dem Betreten steuern Fußgänger*innen auf Attraktionen, wie Lobbys, Infopoints, Toiletten oder Ladeneinheiten zu. Diese Bewegung wird von Barrieren behindert. Im Innenraum handelt es sich hierbei um Wände und geschlossene Räume, Absperrungen, Kunstgegenstände, Wartungsarbeiten und Baustellen im Betrieb.

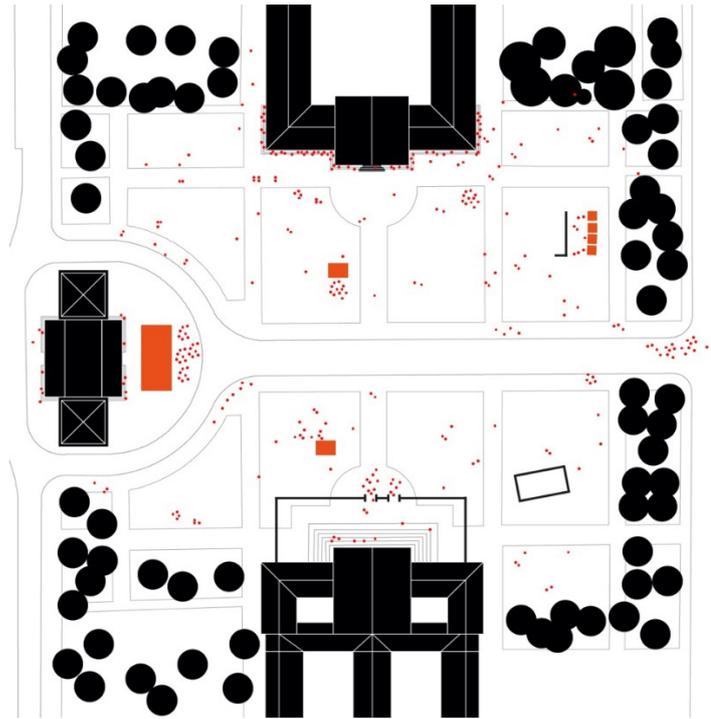


Abbildung 5 Szenario „Outdoor“ – Königsplatz (Quelle: Lehrstuhl für Architekturinformatik)

Szenario 2: Außenraum

Als weiteres Anwendungsszenario dienen öffentliche, variabel genutzte Freiflächen, wie etwa der Königsplatz in München (Abbildung 6). Auf diesen Flächen überlagern sich die Nutzungen angrenzender öffentlicher Gebäude, Großveranstaltungen wie Konzerte und Demonstrationen, sowie temporäre Sperrungen und Baustellen. Bestehende Bereiche müssen kurzfristig für verschiedenartige Anforderungen angepasst werden. Da die räumlichen Anforderungen je nach Veranstaltungsart wechseln, müssen permanent Planungsentscheidungen in Hinblick auf Besucherkomfort und Personensicherheit getroffen werden. Die Integration von Personenstromsimulationen in den Entwurf ermöglicht eine agile Planung, trotz einer Vielzahl von Beteiligten und komplexen Interferenzen zwischen verschiedenen Platznutzungen. Bestehende Bereiche müssen kurzfristig für heterogene Nutzungen adaptiert werden. Da die räumlichen Anforderungen je nach Veranstaltungsart wechseln, müssen permanent Planungsentscheidungen in Hinblick auf Besucherkomfort und Personensicherheit getroffen werden. Die Integration von Personenstromsimulationen in den Entwurf ermöglicht eine agile Planung, trotz einer Vielzahl von Beteiligten und komplexen Interferenzen zwischen verschiedenen Platznutzungen.

Interaktionen: Im Außenraum-Szenario überprüfen Planungsbeteiligte die Anordnung von Funktionselementen innerhalb eines bestehenden Freiraums. Sie verschieben Barrieren und Funktionsobjekte wie z.B. Toilettenhäuschen oder Bühnen in einer vorgegebenen räumlichen Situation.

Insofern müssen schnell geometrische und funktionale Varianten erprobt werden. Großveranstaltungen können zudem als Prozesse aus Einlass, Aufenthalt und Ausgang betrachtet werden, die es aktiv räumlich zu entwerfen gilt. Planer*innen müssen in der Regel den, aus *OpenStreetMap*, geladenen Gebäudebestand nicht anpassen, benötigen aber die Freiheit, Barrieren, Funktionselemente, Eingänge und Zugänge zu ergänzen, frei zu verschieben und zu modifizieren.

Elemente: Im Outdoorszenario bilden angrenzende Erschließungsflächen (Straßen, Bus- und U-bahnhaltestellen) und Gebäudeeingänge die Zugänge zum Entwurfsgebiet. Weniger relevante Anziehungspunkte (Attraktionen) für Fußgänger*innen sind öffentliche Einrichtungen wie Toilettencontainer oder Verkaufsstände. Hauptattraktionen sind beispielsweise Veranstaltungsbühnen. Darüber hinaus bilden Gebäude, Bepflanzungen und Absperrungen (Zäune, Wellenbrecher) Barrieren, welche die Bewegungsfreiheit von Fußgänger*innen einschränken.

4.2 Eingabe

Vor dem Beginn der Personenstromsimulation müssen notwendige Eingabeparameter definiert werden. Im Kontext von Personenstromsimulationen umfassen diese Parameter geometrische Simulationsobjekte (Geometrie, Position, Einstellungen), Agenteneinstellungen (Anzahl, Zielabfolge, Geschwindigkeit, Größe) sowie allgemeine Simulationseinstellungen (Auflösung des Navigationsfelds, dynamische oder statische Berechnung der Navigation etc.). Nutzerinteraktionen müssen korrekt als Eingabeparameter interpretiert und über das Kommunikationsprotokoll an die Simulation übergeben werden.

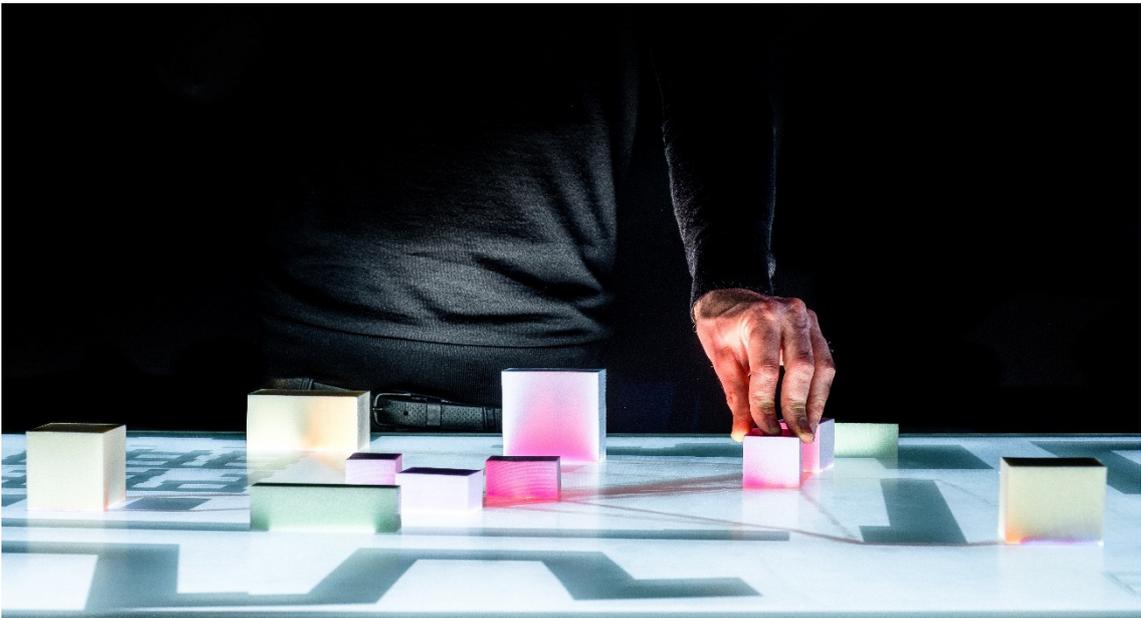


Abbildung 6 Parametereingabe durch intuitive Interaktionen mit physischen Objekten
(Quelle: Lehrstuhl für Architekturinformatik)

Anforderungen

Die untersuchten Anwendungsszenarien beschreiben Entwurfsituationen, in denen heterogene Planungsbeteiligte kollaborativ und simultan mit dem Entwurfsmodell interagieren. Die Eingabe muss somit für alle Entwerfenden gleichermaßen hierarchielos möglich sein. Gleichzeitig muss die Eingabe intuitiv verständlich sein, um Planungsbeteiligte ohne dezidiertes technisches Vorwissen nicht auszuschließen. Die Interaktion mit der Simulation muss mühelos und unaufwändig sein, um den Entwurfsprozess nicht zu unterbrechen – sie darf weder eine Ablenkung noch eine Verzögerung in Entwurfsdiskussionen darstellen.

Darüber hinaus dürfen Umfang und Detailgrad der geforderten Eingabeparameter die verfügbaren Informationen in frühen Entwurfsphasen nicht übersteigen, müssen jedoch eine Grundlage für fundierte Simulationsergebnisse bilden. Insofern mussten Strategien entwickelt werden, um die Eingabe von Parametern und die Definition von Simulationseinstellungen zu vereinfachen und an die geometrischen Interaktionen der *CDP* zu adaptieren.

Lösungsansatz

Ausgehend von diesen Anforderungen wurden vereinfachte intuitive Interaktionsformen entwickelt: einerseits die Wahl eines grundlegenden Simulationsszenarios, andererseits geometrische Entwurfsaktionen. Auf der *CDP* wird die Simulationseingabe mit entwerferischen Interaktionen synchronisiert: Planungsbeteiligte diskutieren anhand eines einfachen Styropormodells verschiedene Entwurfsoptionen und definieren gleichzeitig den Geometrie-Input für die Simulation. Planungsrelevante Elemente werden somit durch physische Objekte kontrolliert. Weitere Objekte können über Handgesten intuitiv gezeichnet werden (Abbildung 6). Dies ermöglicht eine hierarchielose Interaktion mehrerer Akteure mit dem Simulationsmodell – im Gegensatz zu dem restriktiven Interface eines Desktopcomputers. Die Beteiligten einer Planungssituation versammeln sich um das physische Interface und interagieren hierarchielos und simultan mit der Simulation.

Die vereinfachten und geometrischen Nutzereingaben müssen als eindeutige Parameter interpretiert werden. Simulations- und Agenteneinstellungen und die Definition von Simulationsobjekten werden vereinfacht, vorweggenommen und aus den intuitiven Eingaben der *CDP* übersetzt. Diese Strategie forciert einen angemessenen Kompromiss zwischen Benutzerfreundlichkeit und der Präzision der Berechnungsergebnisse. Insofern wurde eine Unterscheidung zwischen **Back-End** und **Front-End** Parametern eingeführt. Während erstere der Nutzerinteraktion vorweggenommen und durch Standardwerte definiert werden, werden letztere unmittelbar von der Nutzer*in modifiziert.

Back-End Parameter

Diese Parameter werden Entwicklerseitig vorab definiert und sind der Nutzer*in nicht unmittelbar zugänglich. Es handelt sich um Simulationsparameter, die in Hinblick auf *das Level of Detail* früher Entwurfsphasen durch Standardwerte festgelegt werden können, ohne qualitative Simulationsergebnisse signifikant zu verfälschen. Häufig können normalverteilte Standardwerte zur Beschreibung des Modells herangezogen werden. Bei spezifischen Abweichungen empfiehlt es sich, diese Werte zu überschreiben. Es handelt sich um die **Agenteneinstellungen**, die Einstellungen zum **Agentenverhalten** und grundlegende **Simulationseinstellungen**.

Agenten Einstellungen repräsentieren Werte für die Gehgeschwindigkeit, Agentengröße und den Platzbedarf der Agenten (Tabelle 1). Für eine nicht spezifizierte Population können Durchschnittswerte übernommen werden.

Tabelle 1 Agenten Einstellungen

<i>Name</i>	<i>Variable Typ</i>	<i>Default</i>	
<i>Min velocity (m/s)</i>	float	0.46	Minimale Geschwindigkeit
<i>Mean velocity (m/s)</i>	float	1.34	Durchschnittsgeschwindigkeit
<i>Max velocity (m/s)</i>	float	1.61	Maximale Geschwindigkeit
<i>Deviation for velocity (m/s)</i>	float	0.26	Abweichung der Geschwindigkeit
<i>Min torso diameter (m)</i>	float	0.42	Minimaler Körperdurchmesser
<i>Max torso diameter (m)</i>	float	0.46	Maximaler Körperdurchmesser
<i>Perception radius (m)</i>	float	2.00	Wahrnehmungsradius

Agentenverhalten spezifiziert den minimalen Abstand zu anderen Agenten. Je höher die Anforderungen an den Komfort gestellt werden, desto größer muss der Abstandsradius zu anderen Agenten in Quellen und Wartezonen sein. Für ein Standardszenario können diese Werte vorweggenommen werden (Tabelle 2).

Tabelle 2 Einstellungen vom Agentenverhalten

<i>Name</i>	<i>Variable Typ</i>	<i>Default</i>	
<i>Comfort distance for origins (m)</i>	float	0.0	Komfortradius bei Entstehung
<i>Comfort distance for queue (m)</i>	float	0.1	Komfortradius in der Warteschlange
<i>Max queue derivation (degrees)</i>	float	30.0	Maximale Länge der Warteschlange
<i>Queue width factor (m)</i>	float	1.0	Breite der Warteschlange
<i>Use social distancing model (m)</i>	float	0.9	Soziales Diszanzierungsmodell

Simulationseinstellungen bilden allgemeine Grundlagen des Simulationsprozesses. Sie spezifizieren die Auflösung des Navigationsfeldes, den betrachteten Zeitraum oder die dynamische/statische Berechnung der Agentennavigation. Diese Einstellungen können mit Hinblick auf das *Level of Detail* früher Entwurfsphasen vorab definiert werde (Tabelle 3).

Tabelle 3 Einstellungen zum Simulationsgebiet

<i>Name</i>	<i>Variable Typ</i>	<i>Default</i>	
<i>Height</i>	float	2.0	<i>Höhe</i>
<i>Elevation</i>	float	0.0	<i>Steigung</i>
<i>Cell discretization</i>	float	0.1	<i>Diskretisierung der Zellen</i>

Front-End Parameter

Front-End Parameter sind Kontext- oder Anwendungsspezifisch und müssen direkt durch die Nutzer*in eingegeben werden. Zu diesen Parametern gehört die Wahl eines **Simulationsszenarios** sowie die Eingabe von **Simulationsobjekten**.

Szenario: Studien zur Integration von Energiesimulationen in frühen Entwurfsphasen entwickelten das Konzept, Entwurfsvarianten in Hinblick auf wenige extreme Benchmark-Szenarios zu überprüfen (Attia et al. 2012). Anstatt Entwurfalternativen mit komplizierten und fallabhängigen Einstellungen zu analysieren, sollen sie in Hinblick auf wenige, kritische Szenarien überprüft werden. Dadurch kann eine Entwurfskonfiguration qualitativ aus verschiedenen relevanten Perspektiven untersucht werden. Zur Überprüfung des Entwurfes auf verschiedene Aspekte von Fußgängerkomfort und -sicherheit kann zwischen drei „Benchmark“-Szenarien gewählt werden: **Eingang** (Agenten betreten das Simulationsgebiet an den Ein-/Ausgängen, besuchen einige Nebenattraktionen, und verlassen das Simulationsgebiet über die Hauptattraktion), **Ausgang** (Agenten betreten das Gebiet über die Hauptattraktion und bewegen sich zu den Ein-/Ausgängen, Nebenattraktionen fungieren als bloße Barrieren) und **Aufenthalt** (Agenten betreten die Simulation über die Ein-/Ausgänge, Besuchen Haupt- und Nebenattraktionen, und verlassen das Gebiet wieder über die Ein-/Ausgänge).

Simulationsobjekte: Entwurfsrelevante Elemente wie Quellen, Ziele und Zwischenziele werden mithilfe von physischen Objekten auf der *CDP* durch die Nutzer*innen modifiziert oder mit einfachen Gesten auf der *CDP* gezeichnet (nur Barrieren). Im Entwurfskontext wird zwischen folgenden Objekttypen unterschieden:

- Zugänge: Haltestellen, Gebäudeeingänge, Aufzüge- und Treppenhäuser, etc.
- Nebenattraktionen: Lobbys, Kartenschalter, Verkaufsstände, Infopoints, etc.
- Hauptattraktionen: Bühnen, Ziel während einer Veranstaltung (z.B. Konferenzraum), etc.
- Barrieren: Außenwände, Innenwände, Tragwerkselemente, Bepflanzung, Zäune etc.

Die Farbe der physischen Objekte bestimmt die Bedeutung der Objekte im Entwurfskontext. Diese wird über den RGB-Kamera der Tiefenkamera der *CDP* bestimmt. Die Bedeutung der Objekte im Simulationskontext hängt von der Wahl des Szenarios ab: In einem Aufenthaltsszenario wird ein Verkaufsstand als Zwischenziel simuliert, das Agenten anzieht, während er in einem Ausgangsszenario nur noch eine Barriere darstellt, die es zu umgehen gilt. Somit werden Eingabeobjekte automatisch als Quellen, Zwischenziele, Hauptziele und Barrieren klassifiziert. Die Bedeutung der Entwurfsobjekte im Kontext der Benchmark-Szenarien ist in Tabelle 4 spezifiziert.

Tabelle 4 Entwurfsobjekte im Kontext der Simulationsszenarien

Objekt:				
Bedeutung im Entwurfskontext:	Zugang	Nebenattraktion	Hauptattraktion	Barriere
Bedeutung im Szenario Eingang:	Quelle	Zwischenziel	Ziel	Barriere
Bedeutung im Szenario Ausgang:	Ziel	Barriere	Quelle	Barriere
Bedeutung im Szenario Aufenthalt:	Quelle & Ziel	Zwischenziel	obligatorisches Zwischenziel	Barriere

Während die Farbe der Objekte ihren Typus bestimmt, definiert deren Höhe ihre Kapazität im Simulationskontext. Somit steuert die Höhe eines Objekts die Agentenzahl, die pro Zeitintervall in einer Quelle erzeugt wird, die zu einem Zwischenziel angezogen wird, die gleichzeitig durch ein Ziel die Simulation verlassen kann. Die Abfolge der Zwischenziele und Hauptziele, welche Agenten absolvieren müssen, wird ebenfalls automatisch generiert um die Nutzer zu entlasten. In Abhängigkeit von dem gewählten Simulationsszenario wird ausgehend von einer Quelle eine Abfolge von Zwischenzielen generiert, die ein Agent passieren muss, bevor er durch ein Ziel die Simulation verlässt.

4.3 Simulation

Ausgehend von den Eingabeparametern berechnet die Simulation die Bewegung und Navigation der Agenten. In einzelnen Simulationsschritten nähern sich die Agenten ausgehend von den Quellen zu ihren Zielen und Zwischenzielen. Da der Simulationsalgorithmus von *crowd:it* im Rahmen dieses Projekts nicht adaptiert wird, gilt es zu untersuchen, wie die Simulation mithilfe der Kommunikationsschnittstelle gesteuert und an den Entwurfsprozess angepasst werden kann.

Anforderungen

Um die Simulation von Personen- und Gruppenbewegungen in dynamische und unvorhersehbare Entwurfsprozesse einzubinden, muss das Ansprechen des Simulationsvorgangs angepasst werden. Statt den Entwurfsprozess an den Simulationsprozess anzupassen, muss die Simulation der Dynamik des Entwerfens folgen. Die Simulation darf den Entwurfsprozess nicht durch passive Berechnungszeiten behindern. Insofern müssen Wege gefunden werden, einerseits die Simulation zu beschleunigen und andererseits Zwischenergebnisse zu verarbeiten. Entwurfsinteraktionen in frühen Phasen können unvorhersehbar und experimentell sein. Geometrische und funktionale Varianten werden in Diskussionen erprobt und angenähert. Daher müssen Entwurfsanpassungen zu jedem Zeitpunkt möglich sein – auch während des Simulationsprozesses. Diese Veränderung der Eingabeparameter hat einen notwendigen Neustart der Simulation zur Folge. Es gilt neue Konzepte zu entwickeln, wie diese Unterbrechung ausgeglichen werden kann, und wie abgebrochene und neu gestartete Simulationen möglichst nahtlos aneinander anknüpfen können.

Lösungsansatz

Um unabhängig von dem Endgerät eine hohe Rechenleistung – und möglichst kurze Rechenzeiten – zu erreichen, wird die Simulation, wie beschrieben, auf einem externen, leistungsfähigen *Server* ausgeführt. Das Kommunikationsprotokoll übermittelt Eingabeparameter an diesen *Server*, initiiert und beendet Simulationsvorgänge und erhält Simulationsergebnisse, sobald diese verfügbar sind. So wird einerseits eine flexible Steuerung der Simulation(en) möglich: Bei einer neuen Simulationseingabe, kann die bestehende Simulation abgebrochen werden, und eine neue Simulation mit angepassten Eingabeparametern gestartet werden. So werden kontinuierliche Nutzerinteraktionen mit dem Entwurfsmodell ermöglicht.

Bei jeder Interaktion starten die Agenten erneut bei den Quellen und eine gleichmäßige Verteilung der Agenten entsteht erst im Laufe des Simulationsprozesses – der Neustart führt also zu einem Bruch, den es auszugleichen gilt. Andererseits wird bei einem Abbruch und Neustart der Simulation die Zielwahl, die Position und die Navigation der Agenten gelöscht – dies hat zum einen den Nachteil, dass diese Parameter neu berechnet werden müssen, und andererseits dass Agenten nach einem Simulationsneustart, nicht am gleichen Ort starten und in völlig verschiedene Richtungen weiterlaufen – was insbesondere nach einer kleinen Entwurfsanpassung unverhältnismäßig verwirrt.

Anpassung des Simulationsvorgangs

Aus diesem Grund wird das klassische, „finite Simulationsmodell“ von *crowd:it* zu einem „kontinuierlichen Simulationsmodell“ (Abbildung 7) angepasst. Eine Methode wird erarbeitet, um Simulationen zu überlagern, um den interaktionsbedingten Neustart zu kompensieren.

Finites Simulationsmodell: Die Anwendung *crowd:it* betrachtet Szenarien mit einer finiten Anzahl von Agenten und einem begrenzten Zeitraum – beispielsweise die Evakuierung von 100 Personen aus einem Gebäude. Die begrenzte Anzahl von Agenten startet an ihnen zugewiesenen Quellen. Von dort aus bewegen sie sich zu ihnen zugewiesenen Zielen und verlassen dort die Simulation (Siehe Abbildung 7). Die Simulation endet, sobald alle Agenten ihr Ziel erreicht haben. Während diese Simulationsmethode für die Bestimmung von absoluten Räumungszeiten die akkurateste Methode darstellt, hat sie den Nachteil, dass sie zunächst lediglich Simulationsinformationen für die Startbereiche bereitstellt und erst zum Ende hin Informationen über den Bereich in der Nähe der Ziele. Ein Neustart der Simulation erfordert zudem eine Platzierung der Agenten an ihren ursprünglichen Startpositionen. Somit muss bei einer Entwurfsanpassung in der Nähe des Ziels die komplette Simulation neu berechnet werden, was dazu führt, dass die Auswirkungen einer Entwurfsanpassung erst spät sichtbar werden. Insofern ist die Vergleichbarkeit von Entwurfsvarianten durch dieses Simulationsmodell begrenzt.

Kontinuierliches Simulationsmodell: Das finite Simulationsmodell wird adaptiert, um kontinuierlich Informationen für das ganze Simulationsgebiet zu erzeugen, durchgängig Interaktionen zuzulassen und die Unterbrechung durch interaktionsbedingte Neustarts zu minimieren. Wie in Abbildung 7 erkennbar ist, durchlaufen in diesem Modell solange Agenten den Simulationsbereich, bis die Simulation gestoppt wird. Agenten betreten den Simulationsraum fortlaufend über die Quellen, passieren die Wartebereiche und verlassen die Simulation über die Ziele (Abbildung 7). Um eine bereits eine anfängliche Verteilung der Agenten zu erzielen, werden zum Simulationsstart bereits einige Agenten zwischen den Quellen und Zielen erzeugt. Somit werden die relevanten Bereiche des Simulationsgebiets durchgehend mit Agenten bevölkert und fortlaufend flächendeckende Ergebnisse generiert.

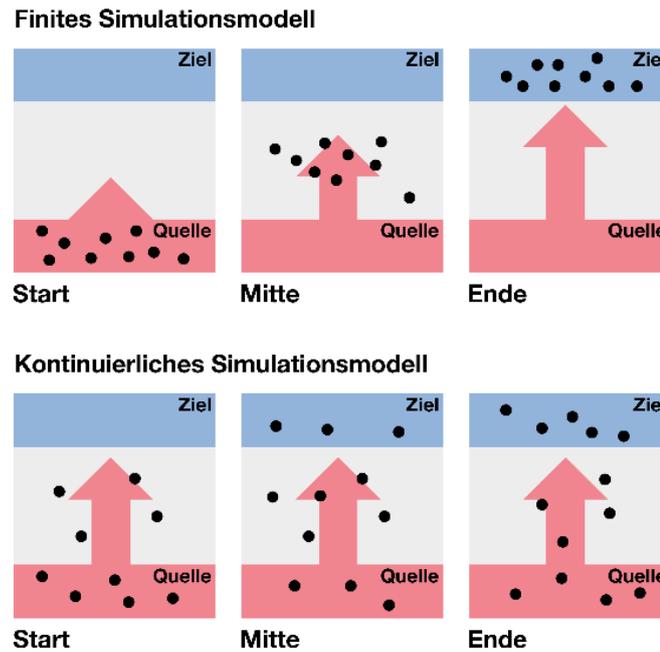


Abbildung 7 Finites vs. kontinuierliches Simulationsmodell (Quelle: Lehrstuhl für Architekturinformatik)

Überblendung von Simulationen

Des Weiteren stellt das Verschwinden und die Erzeugung der Agenten bei jedem interaktionsbedingten Neustart der Simulation eine problematische Unterbrechung dar. Um diese Störung auszugleichen, wird die abgebrochene Simulation mit der neu begonnenen überblendet. Zuvor berechnete Agentenziele und Agentenpositionen werden nach Möglichkeit aus der vorherigen Simulation übernommen. Die Übernahme der Ergebnisse ist in den Tabellen zu „Interaktion Mensch-CDP“ in Abhängigkeit von der jeweiligen Nutzerinteraktion weiter spezifiziert (Tabelle 5 bis Tabelle 7). Agenten werden in der neuen Simulation an den Agentenpositionen der abgebrochene Simulation erzeugt. Die Ziele und Zwischenziele der Agenten werden nach einer Interaktion nur dann angepasst, wenn sie aufgrund der neuen räumlichen Anordnung deutlich ungünstiger zu erreichen sind oder ein angemessenes Alternativziel deutlich leichter zu erreichen ist. So behalten Agenten nach jeder Interaktion nicht nur ihre Position, sondern bewegen sich auch in die gleiche Richtung weiter. Da durch eine neue räumliche Anordnung die Navigationsfeld potenziell nicht mehr mit dem aktuellen räumlichen Aufbau übereinstimmt, muss diese Berechnung nach jeder Interaktion neu ausgeführt werden.

Nach einem Neustart befinden sich die Agenten somit potenziell an Positionen, die sie möglicherweise durch die neuen Eingabeparameter nie erreicht hätten. Da die kontinuierliche Simulation allerdings durchgehend mit neuen Agenten gefüllt wird, während die bestehenden sie wieder verlassen, wird diese Verzerrung mit zunehmender Simulationszeit ausgeglichen und zugunsten einer flüssigeren Nutzerinteraktion in Kauf genommen.

Interaktion: Mensch – CDP

Bei einer Anpassung der Eingabeparameter ist ein Neustart der Simulation notwendig. Abhängig von der Simulation können Agentenpositionen und -Ziele in der neu gestarteten Simulation übernommen werden.

Tabelle 5 Veränderung von Quellen

Quellen Verschieben  Hinzufügen  Entfernen 

Agentenposition

Agenten alt	unverändert	unverändert	unverändert
Agenten neu	spawnen	spawnen	-

Agentenziele

Agenten alt	unverändert	unverändert	unverändert
Agenten neu	Start	Start	-

Tabelle 6 Veränderung von Wartezonen

Wartezonen	Verschieben 	Hinzufügen 	Entfernen 
Agentenposition	unverändert	unverändert	unverändert
Agentenziele	Neuberechnung	Neuberechnung	Neuberechnung

Tabelle 7 Veränderung von Zielen

Ziele	Verschieben 	Hinzufügen 	Entfernen 
Agentenposition	Neuberechnung	Neuberechnung	Neuberechnung
Agentenziele	Neuberechnung	Neuberechnung	Neuberechnung

4.4 Visualisierung

Simulationsergebnisse werden in der Form von XML-Daten ausgegeben und umfassen zunächst lediglich die Position jedes Agenten zu dem jeweiligen Simulationsschritt. Diese Ergebnisse sind für menschliche Nutzer nicht ohne weiteres interpretierbar. Daher müssen sie aufbereitet und im Entwurfskontext sinnvoll dargestellt werden, um die Bewegung von Agenten, räumliche Dichten, oder die Auslastung von Bewegungsflächen zu kommunizieren. Insofern müssen Methoden erforscht werden, um Simulationsergebnisse für die Nutzer*in plausibel anzuzeigen.



**Abbildung 8 Darstellung von Simulationsergebnissen direkt im Entwurfskontext
(Quelle: Lehrstuhl für Architekturinformatik)**

Anforderungen

Wie in den Anwendungsszenarien beschrieben, müssen Simulationsergebnisse für heterogene Planungsbeteiligte intuitiv im Entwurfskontext interpretierbar sein. Darüber hinaus besteht die Frage, welche Informationen in frühen, kontingenten Entwurfsphasen die Entwerfenden unterstützen und wie diese in einem angemessenen Detailgrad dargestellt werden können.

Aus den Nutzungsszenarien ergibt sich die Anforderung, in Echtzeit auf Interaktionen zu reagieren. Um dynamische Entwurfsdiskussionen nicht zu unterbrechen, muss ein Feedback zu neuen Entwurfsvarianten ohne maßgebliche Verzögerung zurückgegeben werden. Nur so können provisorische und experimentelle Entwurfsinteraktionen in der Lösungsfindung einbezogen werden. Bereits kurz nach einer Anpassung müssen erste Ergebnisse bereitgestellt werden, um die Entwurfsadaption zu bewerten. Auch wenn im Kapitel 4.3 beschrieben wurde, wie mithilfe der Einbindung eines leistungsstarken Simulationsservers die Berechnung beschleunigt werden kann, sind finale Simulationsergebnisse nicht umgehend vorhanden. Somit müssen Strategien erforscht werden um unvollständige Zwischenergebnisse ad-hoc zu präsentieren und diese mit zunehmendem Simulationsfortschritt an ein finales Ergebnis anzunähern.

Lösungsansatz

Um die Anforderungen an die Visualisierung zu beantworten müssen die Ergebnisse im Entwurfskontext klar verständlich dargestellt werden, sodass Entwerfende die Auswirkungen von Entwurfsentscheidungen intuitiv einschätzen können. Um dieses Ziel zu erreichen, werden Simulationsergebnisse auf dem Tisch-Bildschirm des *CDP* dargestellt und somit direkt im Simulationsmodell eingeblendet (Abbildung 8). Einerseits müssen Simulationsergebnisse mit einem angemessenem Detailgrad als qualitative Planungshinweise dargestellt werden. Andererseits gilt es die akkuraten Ergebnisse von Multi-Agenten-Simulationen voll für ein Entwurfsfeedback auszuschöpfen. Um dies zu erreichen, wurden verschiedene konventionelle Visualisierungstypen für Personenstromsimulationen untersucht und zu einer auf das Anwendungsszenario abgestimmten „Heatcloud“-Darstellung kombiniert. Darüber hinaus wurden Strategien zur vorläufigen und sich progressiv verdichtenden Darstellung von Zwischenergebnissen erforscht.

Um einem breiten Spektrum an Entwurfsbeteiligten eine relevante Handlungsgrundlage zu bieten, wird das zusätzliche Informationsdisplay genutzt: Hier können zusätzliche Informationen zur Simulation abgebildet werden, ohne vom Entwurf abzulenken. So werden Expert*innen mit Spezialwissen zusätzliche Informationen angeboten.

Typen von Visualisierungen

Um Planungen in Bezug auf Sicherheit und Komfort zu analysieren, ist es relevant, Staus und hohe Personendichten zu erkennen. So können „Bottle-Neck“-Situationen erkannt und entschärft werden. Darüber hinaus prognostiziert die Simulation konkrete Bewegungsmuster und Gruppendynamiken. Mithilfe dieser Bewegungsmuster kann die Platzierung und Auslastung öffentlicher Einrichtungen (wie Toiletten, Shops oder Infopoints) bewertet werden. Außerdem kann die Entstehung von Staus und problematischer Situationen untersucht werden. Diese qualitativen und räumlichen Entwurfsfaktoren dienen in frühen Entwurfsphasen als hilfreiche Hinweise in Diskussionen, während absolute Zahlen (wie akkurate Evakuierungszeiträume) die verfügbaren Input-Parameter übersteigen. Studien über Simulationstools in frühen Entwurfsphasen zufolge, profitieren diese offenen und dynamischen Aushandlungsprozesse weniger von absoluten Werten, als von „Ballpark-Figures“ (Roetzel 2015) mithilfe derer ein Entwurf qualitativ bewertet werden kann. Im Folgenden werden einige grundlegende Visualisierungstypen im Bereich der Fußgängersimulationen beschrieben.

Grundsätzlich lassen sich in Simulationen dargestellte Werte in „rundenweise“ verfügbare und über den gesamten Simulationszeitraum gemittelte Ergebnisse kategorisieren (Kretz 2007). Erstere spiegeln momentane Werte zu einem einzelnen Simulationsschritt wider. Der gesamte Simulationszeitraum kann somit als bewegte Animation dargestellt werden. In dieser Bewegung werden Dynamiken und Veränderungen von Werten über den Simulationszeitraum deutlich. Gemittelte Ergebnisse zeigen hingegen einen Durchschnittswert über den gesamten Simulationszeitraum. Diese Darstellungen haben den Vorzug, momentane Fluktuationen auszugleichen und auf Problemstellungen hinzudeuten, die während des gesamten Simulationszeitraums relevant sind (Kretz 2007). Diese Informationen können nach einer relativen oder absoluten Skala dargestellt werden. Während eine absolute Skala sich auf nicht veränderliche Obergrenzen, wie etwa die sicherheitstechnisch maximal zulässige Personendichte beziehen, beziehen sich aktuelle Skalen auf aktuelle Minima und Maxima, etwa die höchste Agentendichte zu einem bestimmten Simulationszeitpunkt. Die Entwicklung von gut lesbaren Skalen, die intuitiv verständlich anzeigen, ob eine Agentenverdichtung ein gravierendes Problem darstellt oder vernachlässigt werden kann, ist eine weitere Herausforderung im Bereich der Visualisierung.

Typische Visualisierungstechniken für Personenstromsimulationen sind „Animierte Agenten“, „Agenten-Pfade“ und „Heatmaps“, wie sie auch in der klassischen Anwendung *crowd:it* verfügbar sind. Mithilfe dieser Methoden sind verschiedene Aspekte von simulierten Bewegungsmustern und Dichteverteilungen abbildbar.

Animierte Agenten

Die Positionen der Agenten zu dem jeweiligen Simulationsschritt bilden den grundlegenden Output der Simulation. Durch die Abfolge der Simulationsschritte können die Agentenpositionen animiert werden, sodass sie sich analog zur Simulation durch den Simulationsbereich bewegen. Zunächst ermöglicht die konkrete Darstellung der bewegten Agenten einen sehr anschaulichen Einblick in die Personenbewegungen (Abbildung 9). Temporäre Anstauungen und die Entwicklung von Personendichten werden besonders deutlich. Darüber hinaus bietet diese Darstellungsform einen intuitiv verständlichen Einblick für Laien und ungeübte Planungsbeteiligte. Über die Darstellung der Agentenpositionen zu jedem Simulationszeitpunkt hinaus, können auch Informationen über den aktuellen Zustand der Agenten farblich angezeigt werden, beispielsweise deren Geschwindigkeit oder die Agentendichte in ihrer Umgebung. Zudem kann die Anzeigegeschwindigkeit adaptiert werden. Gerade für größere Simulationsbereiche ist eine Echtzeitanzeige für größere Simulationsbereiche nicht schnell genug, um Dynamiken der Personenströme schnell zu bewerten. Wenn die Simulation noch nicht vollständig berechnet ist, kann der jeweils aktuelle Simulationsschritt angezeigt werden.

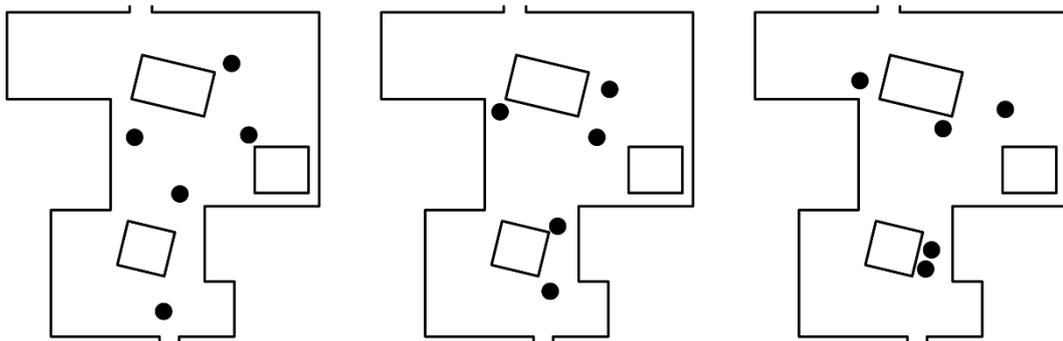


Abbildung 9 Agentenpositionen (in 3 Schritten) (Quelle: Eigene Darstellung)

Agenten-Pfade

Um Personenströme und Gruppendynamiken schnell zu erfassen, ist zudem die Anzeige der Wegpfade der Agenten sinnvoll (Abbildung 10). Sie ermöglicht es, die Bewegungen aller Agenten über den gesamten Simulationszeitraum zu untersuchen und somit globale Bewegungsmuster und Verdichtungen zu erkennen. Dazu wird die Abfolge der Positionen jedes Agenten zu einer Polylinie verbunden – der Laufbahn jedes Agenten. Diese zeigt in welcher Reihenfolge Agenten die einzelnen Zwischenziele passierten und sich mit Bewegungen anderer Agenten kreuzten. Zusätzliche Informationen, wie temporäre Agentendichten, können zudem farblich oder über die Z-Achse einer dreidimensionalen Visualisierung markiert werden und die Anzeige mit weiteren Informationen anreichern.

So kann beispielsweise deutlich gemacht werden, ob viele Agenten eine Engstelle problemlos geordnet nacheinander passieren oder ob es zu einem Simulationszeitpunkt zu problematischen Anstauungen kam. Wenn die Simulation noch nicht vollständig berechnet ist, können die Pfade bis zum aktuellen Simulationsschritt dargestellt werden – wodurch der Eindruck entsteht, dass die mit zunehmender Rechenzeit wachsen.

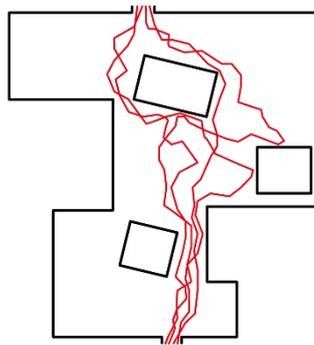


Abbildung 10 Agenten-Pfade (Quelle: Lehrstuhl für Architekturinformatik)

Heatmaps

Dichteverteilungen und Anstauungen im Simulationsraum können zudem durch *Heatmaps* dargestellt werden. Klassische *Heatmaps* bilden den Simulationsraum gerastert ab, und stellen in jedem Rasterfeld ein Berechnungsergebnis als Farbwert dar. Diese Werte können sowohl über den gesamten Simulationszeitraum gemittelt sein wie auch rundenweise variieren.

In seiner Studie zellenbasierter Personenstromsimulationen entwickelt Tobias Kretz verschiedene *Heatmap*-Typen (Kretz 2007): Aus den rundenweise variierenden Agentenpositionen ergibt sich Wert der „Local Density“. Diese Dichte berechnet sich aus der Anzahl der von Agenten besetzten Rasterzellen in der Moorschen Nachbarschaft der Zelle. Dieser zeitlich dynamische Wert kann aufgrund der begrenzten Zellenanzahl der Moorschen Nachbarschaft auf einer absoluten Skala (0 bis 8) dargestellt werden (Kretz 2007).

Die „Occupancy“ einer Zelle hingegen wird über den gesamten Simulationszeitraum gemittelt (Kretz 2007), siehe Abbildung 11. Die Belegung einer Zelle bestimmt sich nach der Anzahl der Bewegungsrunden, in denen eine Zelle besetzt ist im Verhältnis zu der insgesamt simulierten Rundenzahl. Kretz schlägt vor, die Werte zu normalisieren, um so die am stärksten belegten Zellen anzuzeigen (Kretz 2007). Diese Darstellung ist somit relativ zur Simulationszeit, sowie zu aktuellen Minima und Maxima. Um lediglich die relevantesten Verdichtungen anzuzeigen, entwirft Kretz zudem das Konzept der „Significant Congestion“, welches sowohl Agentendichten als auch die Dauer dieser Verdichtungen einbezieht. So sollen nur relevante Anstauungen

angezeigt werden, während zufällige, kurzzeitige Verdichtungen ausgeschlossen werden. Als „Significant Congestion“ werden nur Zellen dargestellt, deren lokale Dichte 4 Personen / qm für mindestens 10 % der Simulationszeit überschreitet (Kretz 2007, 86). Darüber hinaus elaboriert Kretz weitere Darstellungstypen, wie etwa die „Frustration“ von Agenten, die immer dann steigt, wenn sie in ihrer Fortbewegung behindert werden, die „Blockage“ von Agenten durch physische Hindernisse oder Statistiken über den Fortschritt der Evakuierung im Fall eines Ausgangsszenarios (Kretz 2007).

Da die beschriebenen Darstellungsmethoden auf einem räumlich diskretisierten zellulären Automaten basieren, musste deren Übertragbarkeit auf das räumlich kontinuierliche Simulationsmodell *crowd:it* überprüft werden.

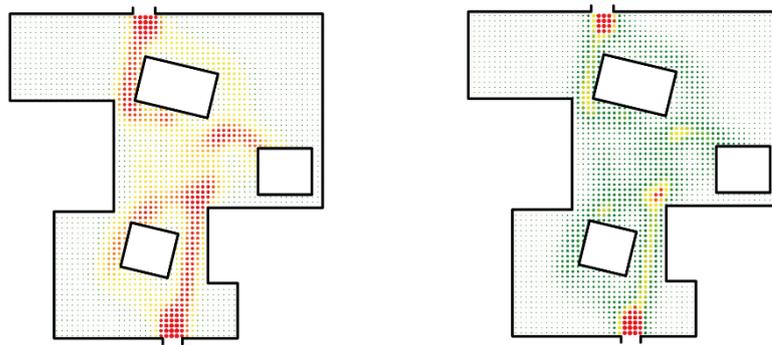


Abbildung 11 „Occupancy“-Heatmap und „Significant Congestion“-Heatmap
(Quelle: Lehrstuhl für Architekturinformatik)

Heatcloud

Nach der Betrachtung unterschiedlicher Visualisierungsmethoden – und deren Vor- und Nachteile – wurde eine Darstellungstechnik entwickelt, die mit Hinblick auf das Anwendungsszenario die beschriebenen Techniken kombiniert (Abbildung 12). Differenzierte *Heatmaps* und statistische Darstellungen sind zwar akkurat, aber schwer zu interpretierbar. Darüber hinaus erscheinen sie ungeeignet, angesichts des qualitativen Detailgrads früher Entwurfsphasen (Attia et al. 2012). Statt absoluten Daten erfordern Diskussionen in offenen und frühen Entwurfsphasen qualitative Hinweise in Hinblick auf die vorgestellten *Benchmark-Szenarien Eingang, Ausgang und Aufenthalt*. Zudem ist die Abbildung lokaler Gruppendynamiken, Bewegungsmuster und Staubildungen ein Vorzug fortgeschrittener Multi-Agenten-Simulationen und sollte dementsprechend dargestellt werden. Um sowohl Informationen über Dichteverteilung, Staubildungen und Bewegungsmuster zu kommunizieren, werden Elemente der *Animierten Agenten*, der *Wegpfade* und *Heatmaps* zu einer räumlich indiskreten *Heatcloud* verknüpft.

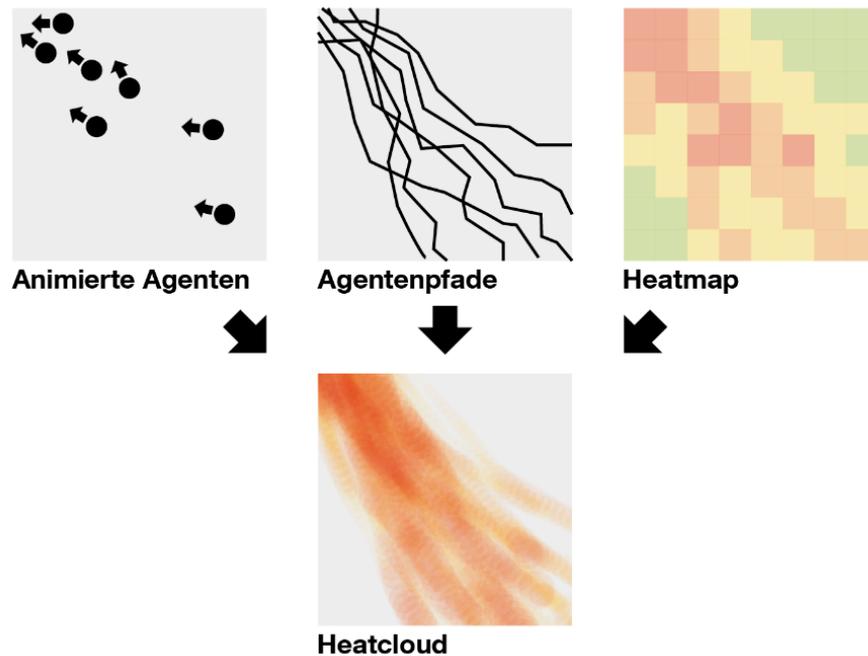


Abbildung 12 Kombination der Darstellungstechniken (Quelle: Lehrstuhl für Architekturinformatik)

Für eine qualitative Bewertung von Personendichten dient das Konzept des „Level of Service“ ein. Diese stammt ursprünglich aus der Verkehrsplanung und wurde zur Beschreibung des Fußgängerverkehrs ausgeweitet (Holl 2016). Dieses Konzept unterscheidet zwischen fünf *Level of Service* in Bezug auf Fußgängerkomfort, Personendichte und Bewegungsgeschwindigkeit. Dieses Prinzip stellt Dichte, Komfort und Geschwindigkeit in „Fundamentaldiagrammen“ als voneinander abhängige Faktoren gegenüber (Holl 2016). Insofern bildet der Indikator *Level of Service* sowohl in sicherheitskritischen Evakuierungsszenarien wie auch in komfortorientierten Aufenthaltsszenarien eine relevante Kenngröße. Der Indikator erlaubt die Verortung von Dichteereignissen auf einer absoluten Skala, die unabhängig von aktuellen Minima und Maxima ist. Dies vereinfacht die Unterscheidung zwischen leichteren Anstauungen und problematischen Verdichtungen. Innerhalb des räumlich nicht diskretisierten Simulationsraums berechnen wir das *Level of Service* durch die Anzahl der Agenten im Radius von einem Meter um den Agenten.

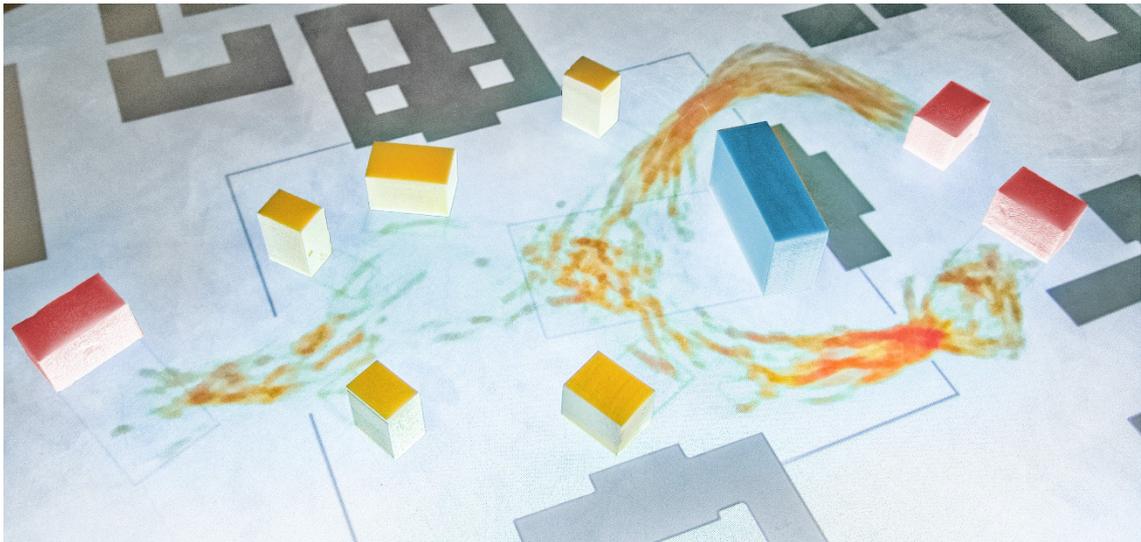


Abbildung 13 Mockup der Heatcloud (Quelle: Lehrstuhl für Architekturinformatik)

Wenn der resultierende Wert eine problematische Dichte indiziert, wird die Position des Agenten als Punkt zur *Heatcloud* hinzugefügt. Diese Punktwolke umfasst somit sämtliche Ereignisse von problematischer Dichte, die während einer Simulation aufgetreten sind. Die Ereignisse werden als Kreise dargestellt, deren Radius und Farbe (nach einer *Heatmap*-Skala) dargestellt werden. Die Kreise werden transparent dargestellt und addieren sich an verdichteten Stellen. So ist es möglich, zwischen leichten und gravierenderen Verdichtungen zu unterscheiden. Darüber hinaus zeigen sich so Bewegungsrichtungen und Dynamiken von Stausituationen – es wird sichtbar wie ein Agent eine Stausituation betritt und diese wieder verlässt. Die Punkte der *Heatcloud* sind mit einem spezifischen Agenten referenziert und im Simulationsbereich lokal zugeordnet. So ist es theoretisch möglich, nach einer Interaktion lediglich die Teile der *Heatcloud* zu überschreiben, die von der Adaption betroffen sind, während andere Bereiche von einer Anpassung unberührt bleiben.

Darstellung vorläufiger Ergebnisse

Entwurfsprozesse und -diskussionen in frühen Entwurfsphasen sind häufig experimentell, iterativ und spontan. Lösungen werden durch zahlreiche Interaktionen angenähert, während verschiedene Entwurfsvarianten diskutiert werden. Simulationsergebnisse können als Hinweise dienen, indem sie eine sukzessive Problemlösung unterstützen (Petersen und Svendsen 2010). Jedoch sind großmaßstäbliche, agentenbasierte Fußgängersimulationen sehr rechenaufwändig. Demnach sind nach einer Nutzerinteraktion mit dem Simulationsmodell, nicht unmittelbar finale Simulationsergebnisse verfügbar. Nichtsdestotrotz muss ein kollaboratives Planungstool in frühen Entwurfsphasen unmittelbar auf Interaktionen reagieren, um den Entwurfsprozess nicht durch passive Wartezeiten zu unterbrechen.

Wie Tests zeigten, simuliert die Anwendung *crowd:it* je nach Umfang des Szenarios annähernd in Echtzeit (1 Sekunde simulierte Zeit entspricht eine Sekunde Rechenzeit). Dies ist für die schnelle Bewertung von Entwurfsveränderungen nicht ausreichend, da die Ergebnisse erst nach mehreren Minuten simulierter Zeit verfügbar wären. Somit ist eine Herausforderung der Visualisierung, umgehend nach einer Interaktion erste vorläufige Ergebnisse anzuzeigen. Mit zunehmendem Simulationsfortschritt können die Darstellung sich an das akkurate Endergebnis annähern. Während die klassische *crowd:it*-Anwendung zunächst die gesamte Simulation berechnet und erst danach die finalen Ergebnisse bereitstellt, sind Zwischenergebnisse nützlich, um nach jeder berechneten Simulationsrunde bereits der aktuelle Stand der Berechnung darzustellen. Mithilfe des kontinuierlichen Simulationsmodells werden sofort erste Ergebnisse für den gesamten Simulationsraum verfügbar (Kapitel 4.3). Mit zunehmendem Rechenfortschritt nähern sich die Zwischenergebnisse immer weiter an das finale Resultat an.

Rundenweise ausgegebene Informationen (wie die Anzeige animierter Agenten) können parallel zum aktuellen Simulationsfortschritt dargestellt werden: Somit kann jede simulierte Runde in Echtzeit dargestellt werden. Analog hierzu werden die Agenten-Pfade oder die *Heatcloud* um Ergebnisse angereichert und erhalten mit dem Rechenfortschritt zunehmende Aussagekraft. Dies führt zu einem intuitiv nachvollziehbaren Wachstum der Simulationsergebnisse (siehe Abbildung 13). Es ist sinnvoll, unvollständige Werte in einer absoluten Skala darzustellen, da aktuelle Minima und Maxima von dem Simulationsfortschritt abhängen und die Darstellung auf einer relativen Skala verzerren würden. Gerade bei großräumlicheren Simulationsszenarien, ist diese Darstellungsmethode in (annäherungsweise) Echtzeit jedoch zu langsam, um die Folgen von Entwurfsentscheidungen unmittelbar einzuschätzen. Da relevante Ergebnisse potenziell erst nach mehreren Minuten verfügbar sein könnten, würde der Entwurfsprozess mit dieser Methode behindert.



**Abbildung 14 Progressive Berechnung der gemittelten Agentendichte über den gesamten Simulationszeitraum
(Quelle: Lehrstuhl für Architekturinformatik)**

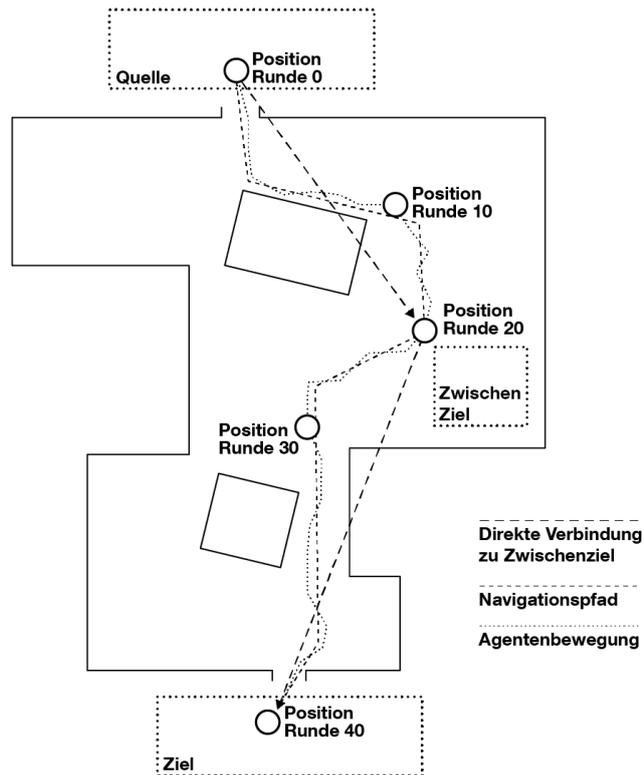


Abbildung 15 Simulationsergebnisse (Quelle: Lehrstuhl für Architekturinformatik)

Um Rechenzeit zu kompensieren werden weitere Berechnungsschritte der Simulation genutzt, namentlich Agentenziele und Navigationsfelder. Diese werden vor der ersten „Bewegungsrunde“ berechnet, und stehen somit umgehend für eine responsive, vorläufige Darstellung zur Verfügung. Mithilfe der vordefinierten Zwischenziele und vorberechneten Navigationsfelder wird eine vorläufige Darstellung der Agentenbewegung ausgegeben. Abbildung 15 zeigt diese vorläufig verfügbaren Informationen: den direkten Weg zum nächsten Zwischenziel, ein approximierter Pfad auf Grundlage des Navigationsfelds und die Zwischenergebnisse der Agentenbewegung. Mithilfe dieser Werte werden die Laufbahnen der Agenten angenähert. Abbildung 16 stellt den aktuellen Aufbau dieser Visualisierungsmethode dar: Ein erstes Feedback wird mithilfe der linearen Verbindung jedes Agenten zu dessen Ziel wird dargestellt. Basierend auf der Basis-Geschwindigkeit der Agenten wird diese Linie in virtuelle Positionen unterteilt, die der Agent in etwa während seinem Weg zum nächsten Zwischenziel passieren wird. Sobald ein Navigationsfeld verfügbar ist, wird dieses genutzt, um die Agentenbewegung weiter anzunähern und genauere virtuelle Positionen zu berechnen. Anschließend werden diese vorläufigen Ergebnisse rundenweise durch die tatsächlich berechneten Positionen der Agenten ersetzt. Somit nähert sich die Darstellung schrittweise dem endgültigen Simulationsergebnis an: Die Information des nächsten Zwischenziels ermöglicht zunächst nur eine sehr unpräzise Vorschau – hier werden lediglich ungefähre Verdichtungen im simulierten Raum sichtbar.

Die Navigationspfade liefern bereits eine präzisere Annäherung an die tatsächlich zu erwartenden Bewegungen der Agenten. Allerdings berücksichtigt diese Vorschau nicht die Geschwindigkeit der Agenten, sowie etwaige Staubildungen oder Kollisionen. Diese Information wird erst durch die letztendliche Berechnung der Simulation bereitgestellt. Durch die Verknüpfung der Punkte der *Heatcloud* mit einzelnen Agenten können sie bereits provisorisch gezeichnet und später aktualisiert werden.

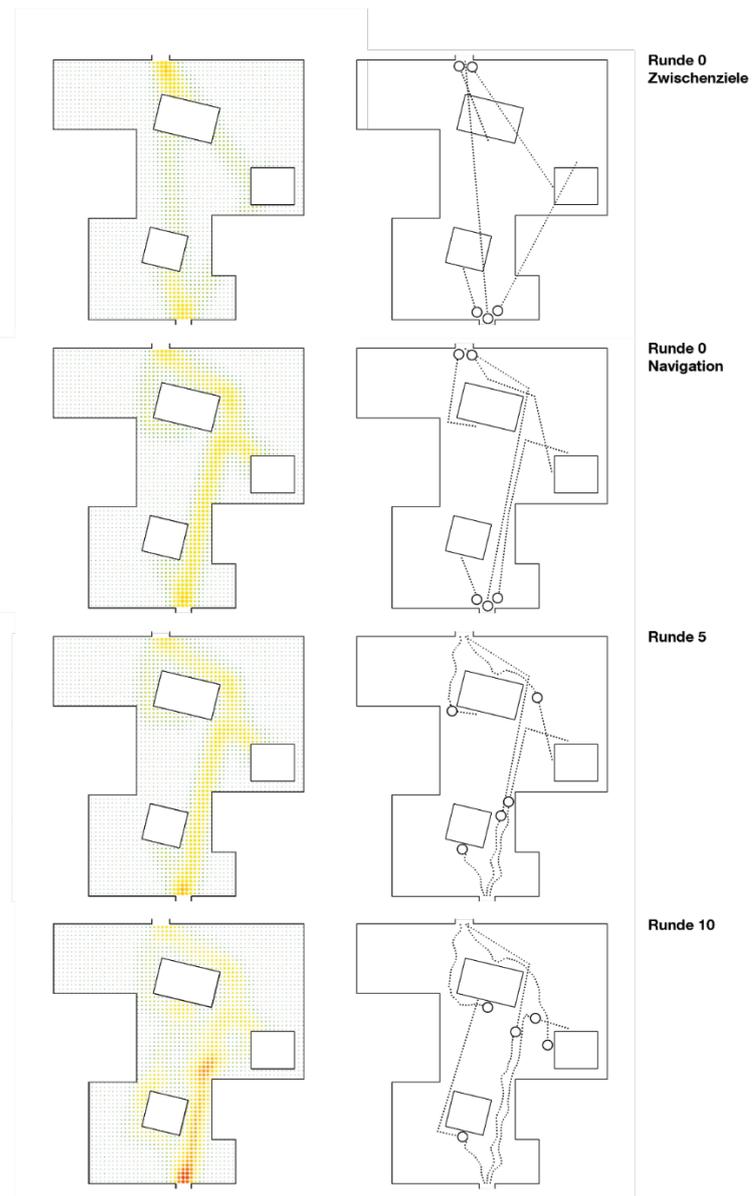


Abbildung 16 Annäherung durch vorläufige Ergebnisse (Quelle: Lehrstuhl für Architekturinformatik)

Bereitstellung zusätzlicher Informationen:

Um zusätzliche Informationen über die aktuelle Simulation abbilden zu können, wird das zusätzliche Infopanel der *CDP* genutzt. So ist es möglich, Informationen unterschiedlicher Komplexität bereitzustellen – während Laien sich hauptsächlich auf die räumliche und intuitiv verständliche Anzeige auf dem Tisch konzentrieren, werden so für Expertengruppen weitere Informationen zu Simulationsberechnungen (Abbildung 17), Agentenverhalten und dem Stand der Simulation ausgegeben.



Abbildung 17 Prototyp des Informationsbildschirms (Quelle: Lehrstuhl für Architekturinformatik)

Bezüglich des **Simulationsfortschrittes** werden folgende Information dargestellt:

- Anzahl der Agenten, die sich aktuell in der Simulation befinden (als Graph)
- Bisher simulierter Zeitraum in Minuten (als Zahl)
- Anzahl der gespawnten Agenten pro Minute (als Zahl)

Bezüglich des **Agentenverhaltens** werden folgenden Informationen dargestellt:

- Anzahl der Agenten, die warten oder sich in einem Stau befinden in Minuten (als Graph)
- Durchschnittliches *Level of Service* (als Zahl)
- Peakdichten in Agenten pro qm (als Zahl)
- Durchschnittliche Wartezeit pro Agent in Minuten (als Zahl)

Zudem zeigt der Bildschirm eine Legende der physischen Objekte im Entwurfskontext und im Simulationskontext (z.B. gelbes Objekt = Attraktion = Zwischenziel). Dies verhilft den Entwurfsteilnehmer*innen zu einem leichteren Verständnis des Simulationsvorgangs (Abbildung 21).



**Abbildung 18 Informationsdisplay als Ergänzung der horizontalen Anzeige
(Quelle: Lehrstuhl für Architekturinformatik)**

5 Technische Umsetzung

Die CDP wurde als Basis für die technische Umsetzung dieser Forschungsarbeit genutzt, siehe Abbildung 19. Nach einer Optimierung und Erweiterung der Codestructur der CDP (I.) wurde ein neues Plugin für die Plattform entwickelt (II.). Die prototypische Implementierung eines *Open-Source*-Kommunikationsprotokolls der *Server-Client*-Architektur ermöglicht den Austausch von Simulationseinstellungen, Geometrie und (Zwischen-) Simulationsergebnissen zwischen der CDP (III.) und dem Simulationsserver (IV.).

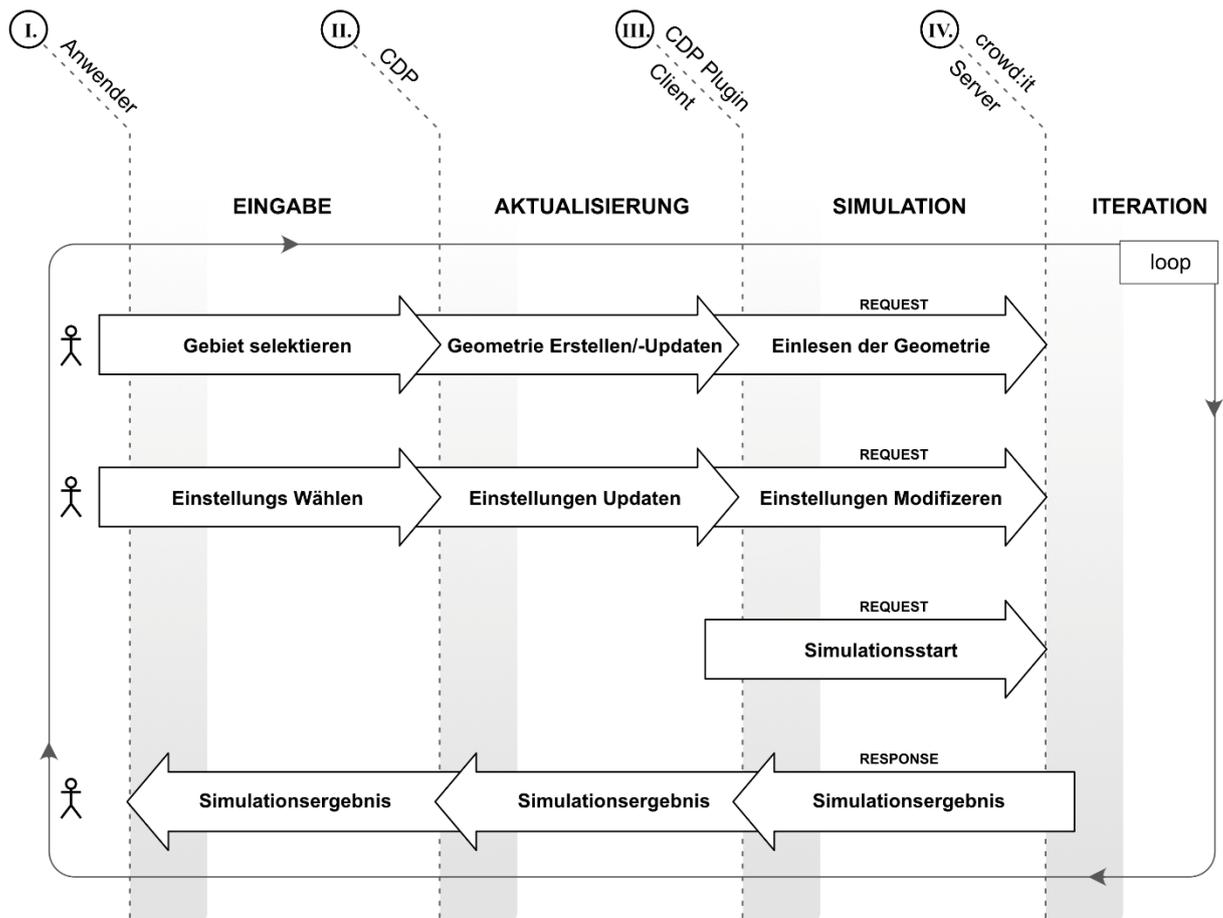


Abbildung 19 Sequenz Diagram (Quelle: Lehrstuhl für Architekturinformatik)

5.1 Technische Grundlagen

a. CDP

Wie bereits beschrieben, baut die Plattform *CDP* auf der Programmiersprache C++ auf. Zur Umsetzung neuer Plugins existiert eine C# Schnittstelle, welche in der Lage ist, auf die Grundfunktionen im C++ Programmcode zuzugreifen. Dies ermöglicht ein objektorientiertes Arbeiten und die Integration vorhandener Bibliotheken (*Dynamic-Link-Libraries*). Die Plattform wurde um einen detailgetreueren Import von Objektdaten aus *OpenStreetMap* erweitert. Dies ermöglicht, dass Gebäudemodelle mehr als einfache Volumina sind, welche als Extrusion von Konturdaten generiert werden. Diese Volumina können nun weitere semantische Informationen wie Eingänge, Dächer, Stockwerke, Verbindungen zwischen Stockwerken, etc. enthalten. Weitere Änderungen an der *CDP* ermöglichen den Plugins einen besseren Zugriff auf die zugrunde liegende C++ Struktur, was zur Erstellung robusterer Benutzeroberflächen, komplexerer Visualisierungen und detaillierterer Informationen über die mit der Plattform verwendeten physischen Objekte führt.

b. Server-Client-Architektur

Für die Kommunikation muss fortwährend ein Austausch zwischen *CDP* und *crowd:it* stattfinden, um ergebnisabhängige Prozesse auszuführen. Dies ist möglich durch eine Verteilung der Rechenprozesse zwischen verschiedenen Rechnern, die miteinander kommunizieren (Bengel 2014). Dafür eignet sich ein *Server-Client-Model*. Der Zweck dieser Softwarearchitektur ist die unkomplizierte Kommunikation zwischen den Plattformen und der Integrierbarkeit des Open-Source-Datenprotokolls an ein beliebiges Entwurfswerkzeug.

Der *Client* fordert den Dienst eines *Servers* an. Dabei können ein oder mehrere *Clients* auf einen *Server* zugreifen. Die einzelnen *Clients* haben keine Informationen über einander, während ein *Server* alle *Clients* zuordnen kann (Bengel 2014). Für den Transfer zwischen *CDP* und *crowd:it* wird ein *Client* und ein *Server* benötigt.

Die Aufgabe eines *Clients* ist, Daten zu empfangen. Daher kann ein *Client* als Konsument betrachtet werden. In diesem Kontext repräsentiert ein *Server* einen Produzenten und sendet Daten an das Netzwerk (Bengel 2014). Der *Server* soll entweder als integrierter Teil eines Simulationstools oder als eigenständige Komponente eingesetzt werden, um mit den Simulationprozessen in *crowd:it* zu interagieren. Für das Forschungsprojekt wurde eine prototypische Version eines *Clients* in dem *Plugin-Framework* der *CDP* implementiert und ein *Server* direkt in die *crowd:it*-Softwarelösung integriert.

c. Kommunikationsprotokoll

Für den Informationsaustausch zwischen dem *Server* und dem *Client* wird ein Kommunikationsprotokoll benötigt, das die Interaktion zwischen den Geräten regelt. *ZeroMQ* ist ein *Open-Source*-Kommunikationsstandard und eignet sich als solcher für das *Open-Source-SDK*. Der Kommunikationsstandard *ZeroMQ* wurde aufgrund der unterstützten Austauschformate, der Verfügbarkeit von Softwarebibliotheken für die plattformabhängigen Programmiersprachen, der einfachen Integration bestehender Projekte und der *Open-Source*-Lizenzierung gewählt. Für den *C#-Client* wird die Software-Bibliothek von *NetMQ* (Akgul 2013) verwendet, während für den *Java-Server* die Bibliothek *JeroMQ* (Bernard 2021) verwendet wird.

ZeroMQ verwendet *Sockets* (Comer und Stevens 1996), um Informationen dezentral zwischen den Geräten zu empfangen und zu senden. Dabei können Informationen über dauerhaft offene Kanäle übertragen werden. Jedes andere Gerät kann ein spezielles Nachrichtensystem verwenden und diese Informationen empfangen. Dieses *Software-Pattern* wird oft als *Subscriber-Publisher-Pattern* bezeichnet. Bei diesem Muster sendet ein Gerät Informationen und Nachrichten (*Publisher*) in verschiedenen Themenkategorien (*Topics*), ohne zu wissen, ob ein anderes Gerät oder eine andere Anwendung die Informationen empfängt. Andere Programme können gesendete Themen empfangen oder abonnieren (*Subscriber*), ohne den Empfänger zu kennen. Mit dieser Art der Kommunikation wurde eine Grundlage für einen Informationsaustausch zwischen beliebigen Entwurfswerkzeugen und Simulationswerkzeugen geschaffen.

Der *ZeroMQ*-Kommunikationsstandard unterstützt zwei Methoden, Nachrichten zu versenden: als Zeichenketten oder als Binärcode. Binärwerte können kompakter gespeichert werden, jedoch ist bei der direkten Übermittlung zwischen unterschiedlichen Programmiersprachen und unterschiedlichen Betriebssystemen besondere Vorsicht geboten, da die Zuordnung der Binärwerte umgekehrt werden kann. Zeichenketten werden von *ZeroMQ* automatisch verarbeitet und sind zudem für den Menschen lesbar und somit leichter zu entschlüsseln. Aus diesem Grund erfolgt die Implementierung der Nachrichtenkommunikation mithilfe von Zeichenketten.

5.2 Informationsaustausch

Um diese Arbeit für weitere Anwendungen und Forschungsprojekte nutzbar zu machen, wurde eine offene Schnittstelle als Basis für das Kommunikationsprotokoll geschaffen. Die Schnittstelle ermöglicht die Integration von *crowd:it* in Entwurfswerkzeuge und ist mit Programmiersprachen wie *C#*, *C++* und *Java* ansprechbar. Sie ermöglicht den Austausch von Simulationsergebnissen, Geometrie- und Eingabeinformationen zwischen dem *Client* und dem *Server* (Abbildung 23).

Die *Publisher*- und *Subscriber*-Klasse wurde sowohl im *Client* als auch im *Server* des *SDK* implementiert, die das Senden und Empfangen von Informationen übernehmen. Weitere Schnittstellen und Klassen erleichtern den Umgang mit den Eingabeinformationen aus den Entwurfswerkzeugen und der Ausgabe des Simulationswerkzeugs.

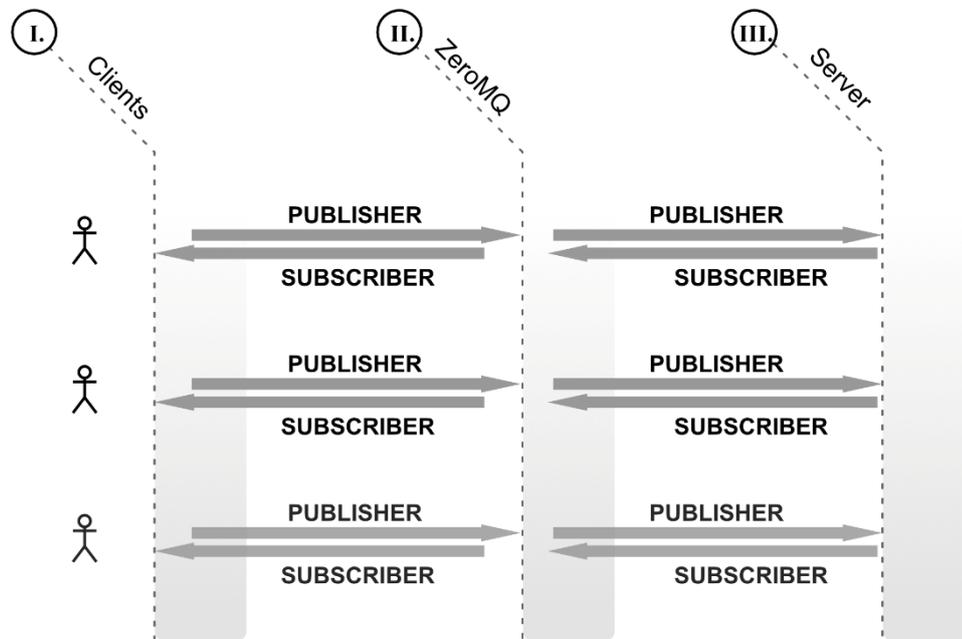


Abbildung 20 Kommunikationsprotokoll (Quelle: Lehrstuhl für Architekturinformatik)

a. Publisher

Die *Publisher*-Klasse beginnt die Kommunikation auf vordefinierten *Sockets* und ist für das Versenden von Nachrichten über diese zuständig. Die Klasse speichert alle zu sendenden Nachrichten in einer Liste und sendet nur die neueste Nachricht in jedem *Topic* einmal pro Sekunde zurück. Dies ist zur Fehlerminimierung bei kleinen Änderungen innerhalb eines kurzen Zeitintervalls notwendig (z. B. beim Verschieben eines Gebäudes). Nach diesem Prozess wird die Liste der älteren Meldungen geleert.

Die *Client*-Implementierung veröffentlicht Informationen zu zwei *Topics*: "geometry" und "settings". Das *Topic* "geometry" übermittelt die Geometriedaten an *crowd:it* als einfache Polylinien. Das *Topic* "settings" veröffentlicht die Einstellungsdatei und wird als *crowd:it*-Datei ausgegeben.

Die *Server*-Implementierung hat ein definiertes *Topic*, auf dem sie veröffentlicht wird: „mainSimulation“. Sie erzeugt jedes Mal ein weiteres *Topic*, wenn Änderungen an der Simulation vorgenommen wurden.

Diese *Topics* sind einfache Zahlenwerte, die nach jedem Start, um ganzzahlige Werte erhöht werden. Die (Zwischen-)Ergebnisse werden nach jedem Start der Simulation veröffentlicht. Das *Topic* „mainSimulation“ dient dazu, jedem zuhörenden *Client* mitzuteilen, auf welchem der generierten *Topics* die aktuellen Simulationsergebnisse veröffentlicht werden.

b. Subscriber

Die *Subscriber*-Klasse hört eingehende Nachrichten an bestimmten *Sockets* ab. Die Klasse empfängt die Nachrichten und leitet diese an den *Client* bzw. *Server* weiter, wo sie weiterverarbeitet werden. Dies geschieht über eine *Delegate*-Funktion für den C#-Client in einer *Observer*-Klasse. Die *Client*-Implementierung empfängt das *Topic* "mainSimulation". Dadurch können weitere *Topics*, z.B. Simulationsergebnissen für den aktuellen Simulationsaufbau, empfangen werden.

c. Support Classes

Um eine nahtlose und benutzerfreundliche Anbindung des Open-Source-SDKs an einerseits die *CDP* und andererseits an *crowd:it* zu unterstützen, sind eine Reihe von Basis-Klassen für den *Client* und den *Server* erforderlich.

IGeometryInputParser-Schnittstellen-Klasse Die Schnittstellenklasse ermöglicht einen standardisierten und zentralisierten Ansatz für die Handhabung verschiedener Geometrietypen der verschiedenen Entwurfswerkzeuge in der *Client*-Implementierung des *SDK*. Sie hat zwei Hauptfunktionen - die erste gibt das aktuell unterstützte Format aus, die zweite führt die eigentliche Konvertierung der Geometrie in einen unterstützten Simulationsgeometrietyp durch. Für den Zweck eines Prototyps wird eine *CDPGeometryInputParsers*-Klasse für diese Schnittstelle umgesetzt und in den *Client* implementiert. So kann die Klasse eine in C# definierte Geometrie verarbeiten und in eine *crowd:it* Geometriedatei konvertieren.

ISettings-Schnittstellen-Klasse Die *ISettings*-Schnittstellen-Klasse dient einem ähnlichen Zweck wie der *IGeometryInputParser*-Klasse. Sie bietet eine übergeordnete Ebene für die allgemeinen Einstellungen einer Simulation. Sie enthält nur eine einzige Funktion, welche die verschiedenen Simulationseinstellungen in einen *String* transformiert. Für die prototypische Implementierung wird dem *Client* eine *CrowditSettings*-Klasse hinzugefügt. In dieser Klasse können Informationen wie Quellenintervalle, Pfade und allgemeine Agentenparameter definiert und automatisch in einen *String*-Wert in der *crowd:it*-Einstellungsdatei umgewandelt werden.

OutputFactory-Klasse Der *Client* empfängt die Simulationsergebnisse als formatierte Zeichenketten. Die *OutputFactory*-Klasse verarbeitet die Informationen auf Basis von Zeichenketten und gibt die Instanz jeder Klassenvariable der Simulation aus. Für die *crowd:it*-Software wurde eine *CrowditOutput*-Klasse generiert, welche einen Wert mit dem aktuellen Zeitschritt und einer Liste aller Agenten, deren Position und Identifikationsnummer und weiteren Variablen, wie z. B. der Bewegungsgeschwindigkeit, enthält.

NetworkThread-Klasse Die *NetworkThread*-Klasse implementiert eine Instanz des *Servers*. Sie verarbeitet die vom *Subscriber* empfangenen Nachrichten am *Server* und verbindet die Ausgabe der Simulationssoftware mit dem *Publisher*. Sobald der *Subscriber* die *NetworkThread*-Klasse über mindestens eine empfangene, gültige Nachricht zum Thema "geometry" und "settings" erhält, erzeugt der *NetworkThread* eine neue Instanz der Simulation und lässt sie im Hintergrund laufen. Sobald die Klasse weitere Aktualisierungen über neue Geometrie- oder Einstellungsänderungen erhält, generiert sie einen neuen Simulationsstart.

5.3 Eingabe

a. Input Parameter

Um physische Objekte als Quellen, Ziele, Barrieren und Hindernisse der Agenten einzulesen, nutzt das Plugin die Objekterkennung der *CDP*. Mithilfe einer *Top-Down*-Tiefenkamera werden die Objekte digital erfasst. Diese kann die Form von physischen Objekten auf dem Tisch einlesen, zusätzlich können bewegte Objekte nachverfolgt werden.

Für die Unterscheidung der Simulationsobjekte wird eine sekundäre Eigenschaft der Tiefenkamera verwendet: Die RGB-Komponente der Kamera wird als ein neues Feature des *CDP* für dieses Forschungsprojekt integriert. Die gemittelte Farbe des physischen Objekts wird nun ebenfalls erfasst und in der Eigenschaftskarte des Objekts als Hexadezimalwert gespeichert. Basierend darauf werden die eingelesenen Simulationsobjekte kategorisiert - als Ein-/Ausgang, Nebenattraktion, Hauptattraktion oder Barriere. Insofern sind Simulationsobjekte durch breite Farbspektren von Rot, Blau, Gelb sowie Grau definiert und in der 2D- und 3D-Ansicht des *CDP* entsprechend dargestellt. In Abhängigkeit von dem Simulationsszenario (Eingang, Ausgang, Aufenthalt) werden diese Objekte für die Simulation als Quellen, Ziele und Wartebereiche interpretiert (siehe 4.2 Eingabeparameter). Die Höhe jedes Objekts definiert dessen jeweilige Kapazität in Bezug auf die Personenstromsimulation – wie viele Agenten in einer Quelle erscheinen, wie viele Agenten einen Wartebereich betreten und wie viele Agenten die Simulation durch ein Ziel gleichzeitig verlassen.

Sobald ein physisches Element als Simulationsobjekt erkannt wurde, erscheint eine weitere Interaktionsmöglichkeit. Dicke Linien erscheinen auf dem Tisch als Umriss der Objekte. Durch die Selektion von Linien können Barrieren in das Simulationsmodell eingefügt werden. Diese verhindert, dass Agenten den Simulationsbereich von dieser Seite aus betreten oder verlassen. Weitere Barrieren können über eine Handgeste definiert werden: Ein langes Tippen auf eine freie Fläche im Simulationsbereich setzt eine Markierung, welche die Bewegung des Fingers verfolgt. Nach dem Absetzen des Fingers, wird dessen Bewegungspfad in eine Barriere umgewandelt.

Wenn mindestens eine Quelle und ein Ziel auf der *CDP* definiert wurden, werden die benötigten Pfade und Spawn-Intervalle automatisch generiert. Für jede Quelle sind die Intervalle von zehn Minuten vordefiniert. In *crowd:it* können Intervalle nicht unendlich wiederholt werden, darum muss zuvor eine feste Anzahl von Intervallen bestimmt werden. Da die Simulation einen ganzen Tag des geplanten Untersuchungsgebietes betrachtet, wird ein Intervall im Abstand von zehn Minuten mehrfach generiert bis ein Tag im Zeitraum von 8 Stunden mit 48 Intervallen befüllt wird.

Wenn das aktuelle Szenario Zwischenziele verwendet, muss eine Menge von Pfaden generiert werden, die diese zwischen Quellen und Zielen einbeziehen. Für jede Quelle erzeugt ein rekursiver Algorithmus mehrere Verzweigungspfade, die auf dem Attraktionsgrad jedes Zwischenziels basieren (durch die Höhe der Eingabeobjekte definiert). Der rekursive Algorithmus nimmt einen Basispfad und eine Liste von noch nicht besuchten Zwischenzielen. Der Algorithmus erzeugt einen semizufälligen Gewichtswert, der wie folgt angegeben wird:

$$weight = \begin{cases} 0 & \text{if } length \leq min \\ step * (length - currAmount) + 0.1 & \text{if } length > min \end{cases}$$

Dabei steht *length* für die Gesamtmenge an Zwischenzielen, *min* für die Mindestmenge an Zwischenzielen, die ein Pfad enthalten muss, *currAmount* für die aktuelle Menge an Zwischenzielen im Pfad und *step* wird angegeben als:

$$step = \frac{0.9}{length - min}$$

Für jede Wartezone, die einen höheren gewichteten Attraktionsgrad hat, wird ein neuer Pfad generiert, der den Basispfad verwendet und ihn um diesen erweitert. Danach wird die rekursive Funktion erneut mit dem modifizierten neuen Pfad und allen verbleibenden Wartezeiten aufgerufen. Wenn es keine Wartezeiten mehr gibt, die eine höhere Anziehungskraft als den Gewichtswert haben, wird das nächstgelegene Hauptziel ausgewählt und dem Pfad hinzugefügt.

Falls das Szenario keine Zwischenziele verwendet, wird für jedes Hauptziel ein direkter Pfad zu jedem Ausgang generiert. Die Verteilung, wie viele Agenten für die jeweilige Quelle dorthin gehen werden, wird anhand der Entfernung zwischen den beiden Bereichen ermittelt.

b. CDP Output to crowd:it

Die *CDP*-Geometrie wird von der *crowd:it*-Simulationssoftware nicht nativ unterstützt. Hierfür ist eine Konvertierung zwischen den beiden Datenformaten erforderlich. Das *crowd:it Floor*-Dateiformat speichert 2D-Konturen aller Quellen, Ziele, Wartebereiche und Barrieren. Da alle Objekte in der *CDP* als 3D Objekte generiert werden, mussten einige Änderungen am *Geometrie*-Export vorgenommen werden. Die

Konvertierung der Geometrie übernimmt eine *CDPGeometryParser*-Klasse der *C#-Client*-Implementierung des *SDK*. Alle Objekte werden mithilfe dieser Klasse für den *crowd:it* Input übersetzt. Dabei wird jedes Objekt überprüft, ob es als Barriere oder als Funktionsobjekt (Quell-, Ziel- oder Wartebereich) kategorisiert wird. Die Funktionsobjekte enthalten die Eigenschaft "wunderzone", um die Funktion der Objekte zu kategorisieren. Um die Geometrie von 3D auf eine 2D-Darstellung zu reduzieren, wird eine Vereinfachung vorgenommen: Die Gebäude werden als deren gespeicherte 2D-Kontur ausgegeben. Der Wert der Objekthöhe wird zurückgesetzt (*y*-Koordinate). Die *crowd:it*-Simulation benötigt eine zweite Datenquelle, in der alle Zusatzinformationen wie Pfade und Intervalle gespeichert sind. Die Klasse *CrowditSettings* des entwickelten *SDK* wird im *CDP*-Plugin verwendet, um alle zusätzlichen Simulationseinstellungen im Plugin zu speichern.

5.4 Simulation

a. Setup, Execution and Output

Simulationswerkzeuge stützen sich auf die Eingabe von Planungsinformationen, wie z.B. Gebäude, Straßen usw., und allgemeine Simulationseinstellungen, wie z.B. die Fußgängergeschwindigkeit oder den Szenariotyp. Um ein besseres Verständnis für den Simulationsvorgang zu erlangen, wurden die Begriffe "geometry" und "settings" für die Simulation festgelegt.

Um ein automatisches Neustarten der Simulation bei Entwurfsanpassungen zu verhindern, wurde eine Möglichkeit geschaffen, mehrere Simulationen parallel laufen zu lassen. Die Simulationsergebnisse werden im Hintergrund berechnet und zu den verschiedenen *Topics* kommuniziert (siehe 5.2.a und 5.2.b).

Die *crowd:it*-Simulation erzeugt für jeden Iterationsschritt eine Liste aller Geschosse in der Simulation. Jeder Ebene wird eine Liste mit allen Agenten auf dieser Ebene zugewiesen. Die Agenten besitzen Informationen ihrer letzten fünf Positionen zu Geschwindigkeit, Richtung und Abstandsradius. Für dieses Forschungsprojekts werden ausschließlich die Informationen zur aktuellen Position der Agenten benötigt. Um diese Informationen zu erhalten, wird eine *Observer*-Klasse implementiert. Nach jedem Simulationsschritt gibt die *Observer*-Klasse die oben genannte Liste aller Ebenen an die *NetworkThread*-Klasse zurück. Die *NetworkThread*-Klasse konvertiert diese Informationen dann in ein *String*-Format mit der Struktur:

```
floorCount; < floorId; pointInTime; agentCount; < agentId; agentPosition; >>
```

Dabei steht *floorCount* für die Anzahl der Ebenen, *floorId* für die ID der Ebene, *pointInTime* für den aktuellen Zeitpunkt in Sekunden, *agentCount* für die Anzahl der Agenten auf der Ebene, *agentId* für die ID der Agenten und *agentPosition* für die Position. Die <>-Klammern stellen ein sich wiederholendes Muster dar. Die *CrowditOutput*-Klasse wird verwendet, um diesen *String* in auswertbare Daten zu übersetzen, die dann vom Plugin für die Analyse und Visualisierung verwendet wird.

b. Umgang mit Simulationsänderungen

Um einen fließenden Übergang zwischen den Simulationsergebnissen zu ermöglichen, wird bei Änderungen an der Geometrie oder den Simulationseinstellungen eine visuelle Überblendung der Ergebnisse durchgeführt. Während die neuen Simulationsergebnisse mit neuen *Topics*, direkt in der Visualisierungsumgebung der *CDP* dargestellt werden, werden die alten Ergebnisse weiterhin an die *CDP* übermittelt. Das *CDP-Plugin* visualisiert die alten Agenten im Betrachtungsgebiet, bis die neuen Agenten mithilfe eines steigenden Transparenzfaktors eingeblendet und überlagert werden. Das Plugin prüft, ob die Agenten aus den alten Ergebnissen mit geometrischen Objekten kollidieren und entfernt diese Agenten aus dem Untersuchungsgebiet.

5.5 Visualisierung

a. Updated Pipelines

Das Ziel der Visualisierung auf der *CDP* ist die intuitive Darstellung von Zwischenergebnissen der *crowd:it*-Simulation.

Durch die Einführung der neuen *Render*- und *Plugin-Pipeline* ist es nun einfacher, neue Objekte und Informationen hinzuzufügen, die in beiden Perspektiven (3D- und 2D-*Top-Down*) dargestellt werden können. Die neue *Render-Pipeline* unterstützt drei Standard-Visualisierungstypen für Objekte; undurchsichtig, transparent und Drahtgitter unter Verwendung einer ordnungsunabhängigen Transparenztechnik. Außerdem kann die Nutzer*in festlegen, ob Objekte in beiden Perspektiven oder nur in einer dargestellt werden soll. Auf diese Weise ist es möglich, jedes Objekt und jede Information abhängig von der Dimension (3D vs. 2D) darzustellen. Bei Verwendung der standardmäßigen Anzeige werden die Bilder nur in der *Top-Down* Sicht dargestellt.

Eine weitere Verbesserung der *Plugin-Pipeline* ermöglicht nun die Darstellung von *Plugin*-erstellten *UI*-Elementen. Der Monitor, der üblicherweise zur Darstellung von 3D-Objekten genutzt wird, kann nun als Projektionsfläche für verschiedene *UI*-Elemente und Komponenten verwendet werden, um zusätzliche Informationen zur Simulation zu präsentieren (z.B. die aktuelle Anzahl an Agenten, die durchschnittliche Wartezeit, usw.). Die unterstützten *UI*-Elemente sind Teil der Interface Bibliothek in Qt (The Qt Company 2021). Sie bietet eine große Auswahl an interaktiven *UI*-Elementen, wie Schaltflächen und Schieberegler, aber auch Elemente, die sich nur auf die Darstellung von Informationen konzentrieren, wie Kreisdiagramme und Graphen.

b. Implementierung

Zur Umsetzung der in 4.4 vorgestellten Konzepte benötigt das *CDP*-Plugin nur die von der *CrowditOutput*-Klasse bereitgestellten Informationen. Wie bereits beschrieben, wird jeder Zwischenschritt der Simulation direkt über die Schnittstelle kommuniziert.

Die Agenten werden in beiden Perspektiven als rote Kugeln visualisiert und ihre Position wird bei jedem neuen Ergebnis aktualisiert. Wenn ein Agent im aktuellen Berechnungsschritt nicht mehr vorhanden ist, wird er aus der Visualisierung entfernt.

Zur Visualisierung der in 4.4 vorgeschlagenen *Heatcloud* wird ein *Transparenz-Rendering* verwendet. Für jeden Agenten wird in jedem Simulationsschritt geprüft, ob im Radius von einem Meter weitere Agenten existieren. Eine Farbskala bildet die Anzahl der Agenten (Agentendichte) innerhalb dieses Radius ab. Der Minimalwert – ein Agent – wird gelb dargestellt, der Maximalwert – 6 Personen – rot. Zwischenwerte werden interpoliert. Um jeden Agenten (der mindestens einen anderen Agenten um sich hat) wird ein Kreis mit dem Radius von 1 Meter in der ermittelten Farbe erzeugt. Sobald alle Kreise für den aktuellen Simulationsschritt erzeugt wurden, werden sie an die *Rendering-Pipeline* weitergeleitet. Danach erhalten alle Kreise, die bis zu diesem Zeitpunkt hinzugefügt wurden, einen aktualisierten Transparenzwert, der $1 / \text{Anzahl der Schritte}$ beträgt (Transparenzwerte reichen von 0 für vollständig durchsichtig bis 1 für vollständig undurchsichtig).

6 Evaluierung

Um die vorgestellte automatisierte Simulationsmethode aus Kapitel 3 zu evaluieren, wird diese einem konventionellen Entwurfs- und Simulationsprozess gegenübergestellt. Dabei wird jeder Arbeitsschritt zeitlich dokumentiert und mithilfe von Zahlenwerte abgeglichen. Der konventionellen Prozess bildet die Planungspraxis ab. Jeder Akteur generiert Inhalt für seine Leistungsphase. Nach erfolgreicher Bearbeitung findet ein Datenaustausch zwischen den Akteuren statt und der nächste Akteur kann den Arbeitsablauf fortführen. Der automatisierte Prozess bietet die Möglichkeit, Arbeitsschritte (z.B. Entwurf und Simulation) zu vereinen. Expert*innen vor Ort entscheiden gemeinsam. Sie sind in der Lage, Ergebnisse valide zu interpretieren. Dieser Prozess bietet die Möglichkeit, aufkommende Fragestellungen während des Entwurfsprozesses im Expertenkreis zu diskutieren.

Für die Evaluierung wurde eine fiktive Entwurfsaufgabe erstellt: Die Planung eines Festivalgeländes am Königsplatz unter Berücksichtigung von Personenströmen. In dieser Betrachtung ist der Fokus auf die Prozessqualität gerichtet. Für diese Untersuchung wurde das Einlasszenario gewählt. Es wurde versucht, die erforderlichen Raumfunktionen für ein Festival realitätsnah abzubilden und anschließend mithilfe von Simulationen auf Sicherheit und Komfort zu testen.

6.1 Konventioneller Prozess

Datenimport: Um auf die gleiche Datenbasis wie *CDP* zuzugreifen, wurde ein Modellausschnitt:

<https://www.openstreetmap.org/> gewählt und als lokale Datei abgespeichert. Der Standort (Königsplatz) ist aus Tabelle 8 Standort München zu entnehmen. Die Standortdaten werden für die Gegenüberstellung beider Methoden benötigt.

Tabelle 8 Standort München

<i>Name</i>	<i>Standort</i>	<i>LON</i>	<i>LAT</i>
<i>Oben, rechts</i>	<i>München</i>	<i>11.5718858900304</i>	<i>48.1485761158618</i>
<i>Unten, links</i>	<i>München</i>	<i>11.5588373028103</i>	<i>48.1434141365506</i>

Anschließend wurden die *OSM*-Daten exportiert (Abbildung 21).

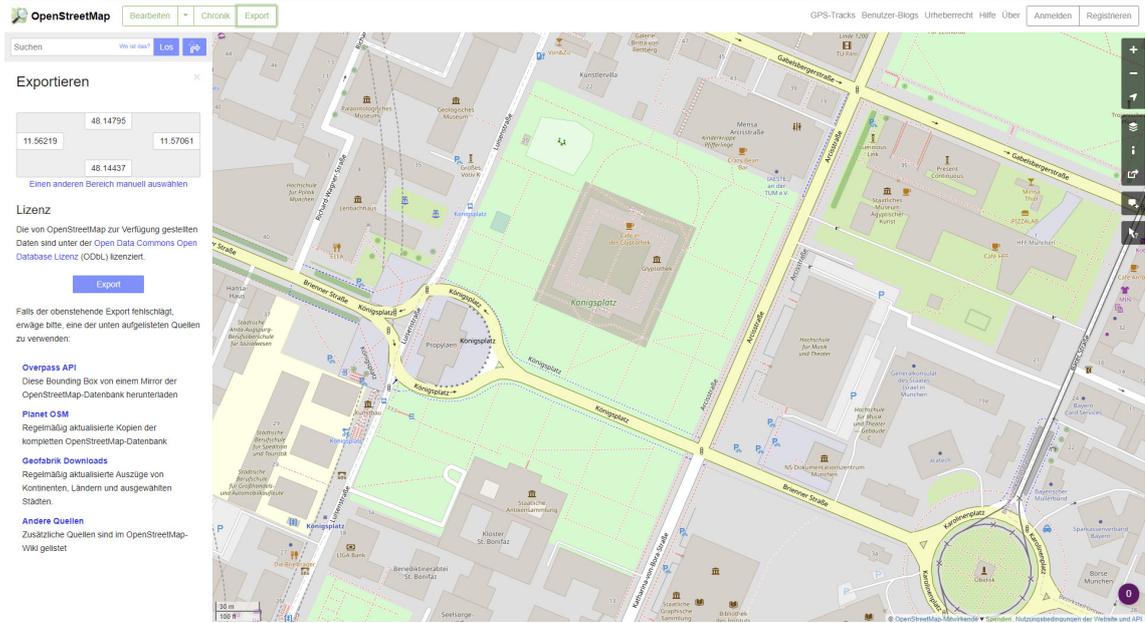


Abbildung 21 Königplatz, OSM Daten Export (Screenshot: Lehrstuhl für Architekturinformatik)

In einer OSM Datei sind geometrische Daten für Straßen, Schiene, Flüsse, Freiflächen und Gebäude gespeichert. Um gezielte Informationen aus der OSM-Datei zu filtern, hat sich das kostenlose Elk-Plugin (<https://www.food4rhino.com/app/elk>) in Rhinoceros 6 und Grasshopper 3D bewährt (Abbildung 22).

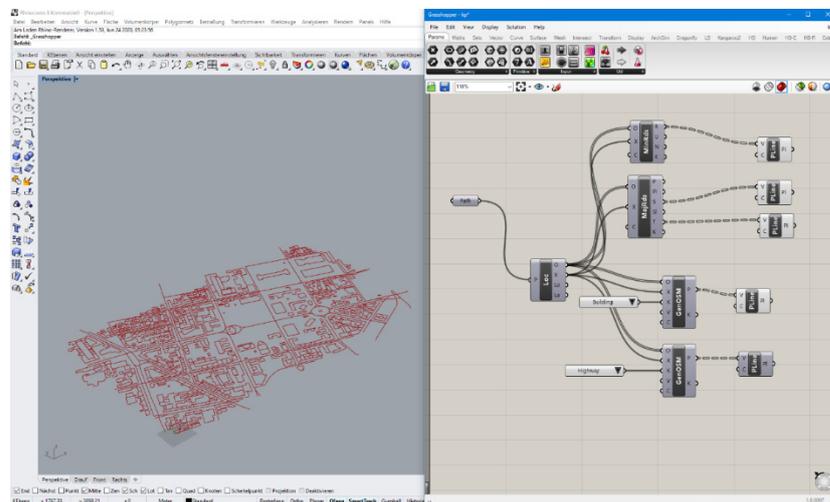
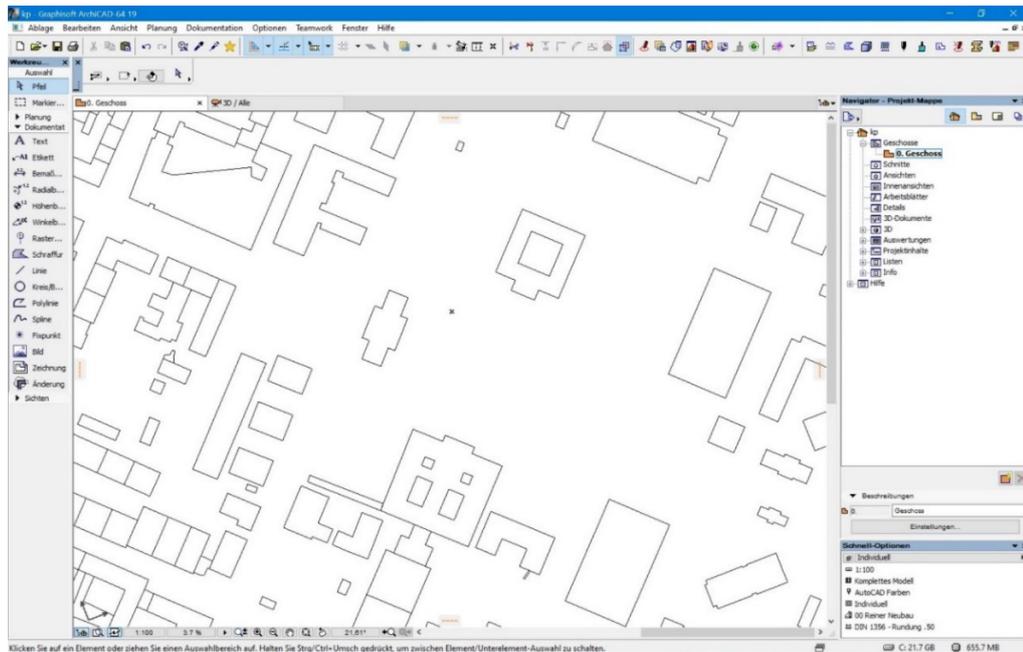


Abbildung 22 Elk-Plugin Schaltung (Rhinoceros 6, Grasshopper 3D) (Screenshot: Lehrstuhl für Architekturinformatik)

OSM ist ein freies und kostenloses Datenformat. Je nach Region oder Stadt sind die Daten unterschiedlich gepflegt. Die Vermessungsdaten der Gebäude sind in weitgehend vorhanden und kongruieren mit den Daten von Stadtverwaltung und Behörden. Diese Datenpunkte werden im *Elk*-Plugin verbunden und können als Polygonlinien exportiert werden. Spezifische Gebäude (wie Museen) können mit dem *Elk*-Plugin nicht gefiltert und erstellt werden, folglich wurde das Planungsgebiet manuell nachbearbeitet.



**Abbildung 23 Ergänzen von fehlenden Gebäuden mithilfe von ArchiCAD
(Screenshot: Lehrstuhl für Architekturinformatik)**

Nachbearbeitung: Das Planungsgebiet wurde in ArchiCAD mithilfe einer DXF-Datei geladen. ArchiCAD ermöglicht (s. Abbildung 23) die einfache Bearbeitung von Zeichnungsdateien und beinhaltet erweiterte (Fang-) Funktionen. Bei dem softwareübergreifenden Datenaustausch wurde zuerst der Maßstab mithilfe der Abmessung von Gebäudekanten überprüft. Die fehlenden Gebäude wurden manuell in Google Maps abgemessen und in ArchiCAD übertragen.

Tabelle 9 Funktionsprogramm

Name	Type	Länge	Breite	Fläche	Agenten	Waiting
<i>Festivalfläche</i>	float	165 m	114 m	18810 m ²	640	s
<i>Eingang 1</i>	float	10 m	10 m	100 m ²	80	900 s
<i>Eingang 2</i>	float	10 m	10 m	100 m ²	80	900 s
<i>Ausgang 1</i>	float	10 m	10 m	100 m ²	∞	∞ s
<i>Ausgang 2</i>	float	10 m	10 m	100 m ²	∞	∞ s
<i>Gate</i>	float	14 m	5 m	70 m ³	20	10 s
<i>Wellenbrecher 1 (Bühne 1)</i>	float	42 m	14 m	588 m ²	300	1800 s
<i>Wellenbrecher 2 (Bühne 1)</i>	float	42 m	23 m	966 m ²	200	1800 s
<i>Zuschauer (Bühne 2)</i>	float	28 m	20 m	560 m ²	150	1800 s
<i>Sanitäter</i>	float	15 m	15 m	225 m ²	10	900 s
<i>Presse/ Crew/ Security</i>	float	15 m	15 m	225 m ²	0	0 s
<i>WC</i>	float	5 m	20 m	100 m ²	15	180 s
<i>Merchandise</i>	float	5 m	20 m	100 m ²	2	180 s
<i>Party-Zelt</i>	float	15 m	45 m	675 m ²	300	1800 s
<i>Bar</i>	float	20 m	10 m	200 m ²	1	180 s
<i>Getränke 1</i>	float	5 m	10 m	50 m ²	1	180 s
<i>Getränke 2</i>	float	5 m	10 m	50 m ²	1	180 s
<i>Getränke 3</i>	float	5 m	10 m	50 m ²	1	180 s
<i>Essen 1</i>	float	5 m	10 m	50 m ²	1	180 s
<i>Essen 2</i>	float	5 m	10 m	50 m ²	1	180 s

Entwerfen: Für das Festivalgelände am Königsplatz wurden die Freiflächen vor den Museen gewählt. Der Platz ist weitgehend versiegelt und wird durch die Propyläen, die Museen und Bäume begrenzt. An der Ostseite des Platzes (Arcisstraße) sind die Ein- bzw. Ausgänge. Objekte wie Zäune und Bühnen wurden auf einem separaten Layer mithilfe von Polygonlinien konstruiert. Alle Objekte auf diesem Layer können von den Agenten nicht betreten werden. Wartebereiche, wie Bars, Getränke- und Essensstände wurden auf dem *crowd:it* Layer erstellt. Diese Objekte können in den Simulationseinstellungen im Detail beschrieben werden. Für das Festival wurde ein Funktionsprogramm erstellt. Wie in Kapitel 3.1 dargestellt, werden Elemente wie Quellen, Ziele, Wartebereiche und Barrieren klassifiziert. Die Elemente des Entwurfszenarios sind in Tabelle 9 aufgelistet. Jedem Entwurfsobjekt wird eine Fläche, eine maximale Kapazität an Agenten, die es gleichzeitig betreten können und eine Aufenthaltsdauer für Agenten zugewiesen. Die Entwurfsobjekte aus Tabelle 9 wurden beispielhaft am Königsplatz (Festivalgelände) angeordnet (Abbildung 24). Anschließend wurde das Modell als DXF-Format exportiert. Bei dieser manuellen Bearbeitung der Prozessschritte ergeben sich Defizite bei Änderungswünschen: Eine Neuordnung der Entwurfsprojekte hat einen wiederkehrenden Datenexport und Neuaufbau des Simulationsmodells zur Folge.

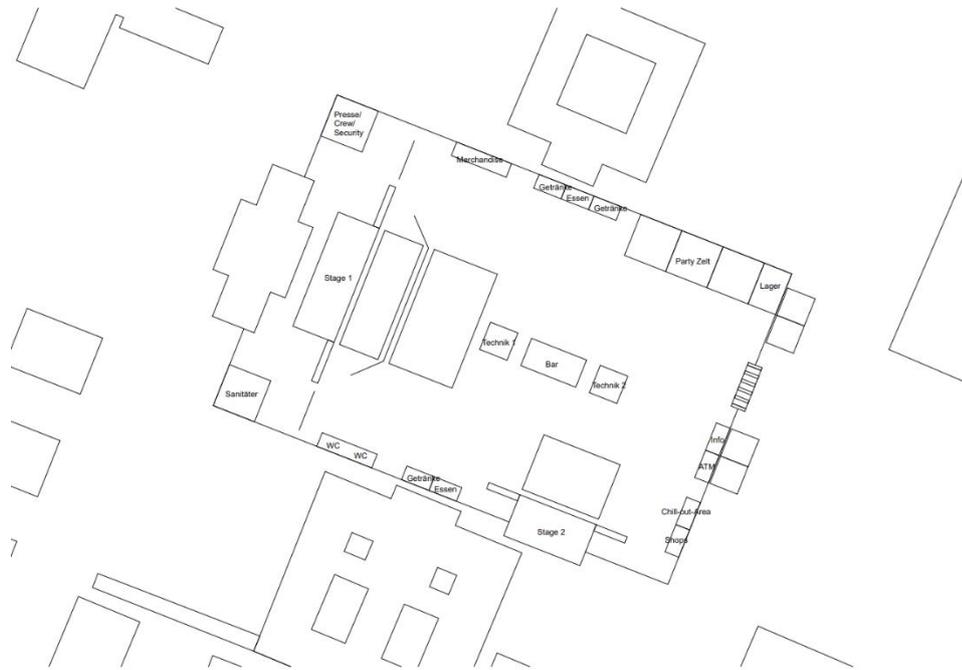


Abbildung 24 Entwurf eines Festivalgeländes am Königsplatz (Screenshot: Lehrstuhl für Architekturinformatik)

Simulation: Im nächsten Schritt wird das Simulationsmodell aufgebaut. Die Voreinstellungen in der *crowd:it* Software wurden angepasst, um die Simulationsdauer – bei höchstmöglicher Genauigkeit – zu reduzieren. Tabelle 5 bis Tabelle 7 zeigen Änderungen der Einstellungen zur Agentengeometrie, zum Agentenverhalten und zum Untersuchungsgebiet.

Tabelle 10 Agenten Einstellungen

<i>Name</i>	<i>Variable Typ</i>	<i>Wert</i>	<i>Default</i>
<i>Min velocity (m/s)</i>	float	0.46	0.46
<i>Mean velocity (m/s)</i>	float	1.34	1.34
<i>Max velocity (m/s)</i>	float	1.61	1.61
<i>Deviation for velocity (m/s)</i>	float	0.26	0.26
<i>Min torso diameter (m)</i>	float	0.42	0.42
<i>Max torso diameter (m)</i>	float	0.46	0.46
<i>Perception radius (m)</i>	float	2.00	2.00

* Deutsche Übersetzung in Tabelle 1

Als Agenten Einstellungen (Tabelle 9) wurden die Standardwerte von *crowd:it* übernommen. Diese Werte bilden das Agentenverhalten realitätsnah ab und haben keinen Einfluss auf die Dauer der Simulation.

Tabelle 11 Einstellung zum Agentenverhalten

<i>Name</i>	<i>Variable Typ</i>	<i>Wert</i>	<i>Default</i>
<i>Comfort distance for origins (m)</i>	float	0.2	0.0
<i>Comfort distance for queue (m)</i>	float	0.1	0.1
<i>Max queue derivation (degrees)</i>	float	30.0	30.0
<i>Queue width factor (m)</i>	float	1.0	1.0
<i>Use social distancing model (m)</i>	float	1.5	0.9

* Deutsche Übersetzung in Tabelle 2

Das menschliche Verhalten hat sich im Jahr 2020 in öffentlichen Räumen aufgrund der COVID-19 Krise verändert. *Accurate* hat auf diese Entwicklung reagiert und einen „Social Distancing“ Modus eingeführt. Kritische Bereiche wie Eingänge können vorab geprüft und der Platzbedarf für Veranstaltungen ermittelt werden. Abstandsunterschreitungen und Schlangenbildung an Engstellen können mit diesem Modus ebenfalls betrachtet werden. Zunächst wurde der Komfortradius auf 1.5 Meter erhöht (Tabelle 11), um das Ansteckungsrisiko durch Viren bei öffentlichen Veranstaltungen zu reduzieren. Eine Änderung dieser Parameter führt zu einer akkuraten Einschätzung des Agentenverhaltens und hat keinen Einfluss auf die Simulationsdauer.

Tabelle 12 Einstellungen zum Untersuchungsgebiet

<i>Name</i>	<i>Variable Typ</i>	<i>Wert</i>	<i>Default</i>
<i>Height</i>	float	3.0	2.0
<i>Elevation</i>	float	0.0	0.0
<i>Cell discretization</i>	float	0.25	0.1
<i>Simulation duration</i>	float	120	5
<i>Simulation Time</i>	float	6.2	-

* Deutsche Übersetzung in Tabelle 3

Die Simulation unterteilt das Planungsgebiet in diskrete Zellen. Die Zellengröße bestimmt die Auflösung im Untersuchungsgebiet und somit die Genauigkeit der Simulation. Eine Veränderung der „Cell Discretization“ von 0.1 Meter auf 0.25 Meter führt zu einer Reduzierung der Simulationsdauer von bis zu 75% (Tabelle 12).



Abbildung 25 Aufbau eines Simulationsmodells in *crowd:it*
Darstellung der Aufenthaltsszenarien (Screenshot: Lehrstuhl für Architekturinformatik)

Die Elemente im *crowd:it* Layer können als Quellen, Ziele oder als Wartezonen definiert werden (Abbildung 25). Elemente auf allen anderen Layern werden als Barrieren erkannt. Neben dem Eingang befinden sich jeweils zwei Quellen und Ziele. Alle anderen Elemente im *crowd:it* Layer sind Wartezonen und wurden mit den Werten aus Tabelle 5 beschrieben. Je nach Auslastung der Wartezonen sollen sich Agenten über das Festivalgelände verteilen. Dafür wurden manuell mehrere Pfade generiert.

6.2 Automatisierter Prozess

Datenimport: Beim Datenimport wurde auf die gleiche Datenbasis wie beim konventionellen Prozess zurückgegriffen. Um das Planungsgebiet zu erstellen, muss zuerst Ausschnitt in *CDP* gewählt werden. Anschließend können die *OSM*-Daten geladen werden.

Nachbearbeitung: Beim Import entfallen ein Nachbearbeiten des Datenmaterials, eine Überprüfung des Maßstabs und der Datenaustausch zwischen Zeichenprogrammen, da alle Geometriedaten nativ von *OSM* zu *CDP* übertragen werden.

Entwerfen: Der Prozess des Entwerfens in *CDP* unterscheidet sich durch eine kollaborative Arbeitsweise. Die Akteure sind in der Lage in frühen Entwurfsphasen Simulationstools in die Lösungsfindung zu integrieren. Für die Vorleistung der Planung und Festlegung des Raumprogramms, kann die gleiche Bearbeitungszeitraum wie für den konventionellen Prozess angesetzt werden. Die Zuordnung von Quellen, Wartebereiche, Ziele und Barrieren erfolgt automatisch mithilfe vordefinierter farbiger Klötzchen (s. Tabelle 4). Für die Erstellung des Funktionsprogramms wurde ebenfalls auf Tabelle 9 zurückgegriffen.

Simulation: Der Aufbau des Simulationsmodells findet automatisiert statt. Für alle Voreinstellungen wurden Parameter im Modell hinterlegt. Wie beim konventionellen Prozess können einzelne Inputparameter durch die Nutzer*in angepasst werden. Das Simulationsergebnisse werden in Echtzeit von einem Server über die Schnittstelle auf die *CDP* zurückgegeben

6.3 Diskussion

Zunächst werdend die Ergebnisse der beiden Prozesse quantitativ verglichen (Bearbeitungszeit), um anschließend die aufgedgliederten Prozesse (Datenimport, Nachbearbeitung, Entwerfen, Simulation) qualitativ zu bewerten. Anschließend werden die Ergebnisse disuktirt.

6.4 Quantitativer Vergleich

In Abbildung 26 ist die Dauer der einzelnen Prozessschritte dargestellt. Die absolute Bearbeitungszeit des konventionellen Prozesses beträgt 240 Minuten, die des automatisierten Prozesses in *CDP* beträgt lediglich 1,17 Minuten. Das entspricht einer Verringerung der Bearbeitungszeit um 20.571%. Wobei ein Großteil der Bearbeitungszeit beim konventionellen Prozess auf die Datenaufbereitung, Datenkonvertierung und die Erstellung des Simulationsmodells entfällt. Die vorgelagerten Prozesse der Simulation konnten in *CDP* automatisiert werden, wodurch das Modell in weniger als 1 Minute zum Simulationsserver gesendet werden kann. Die Veränderung der Gittergröße des Untersuchungsgebiet konnte die Rechenzeit der Simulation um 75% verringern, wodurch die Simulationszeiten je nach Modellgröße von 1 bis 300 Sekunden varriieren.

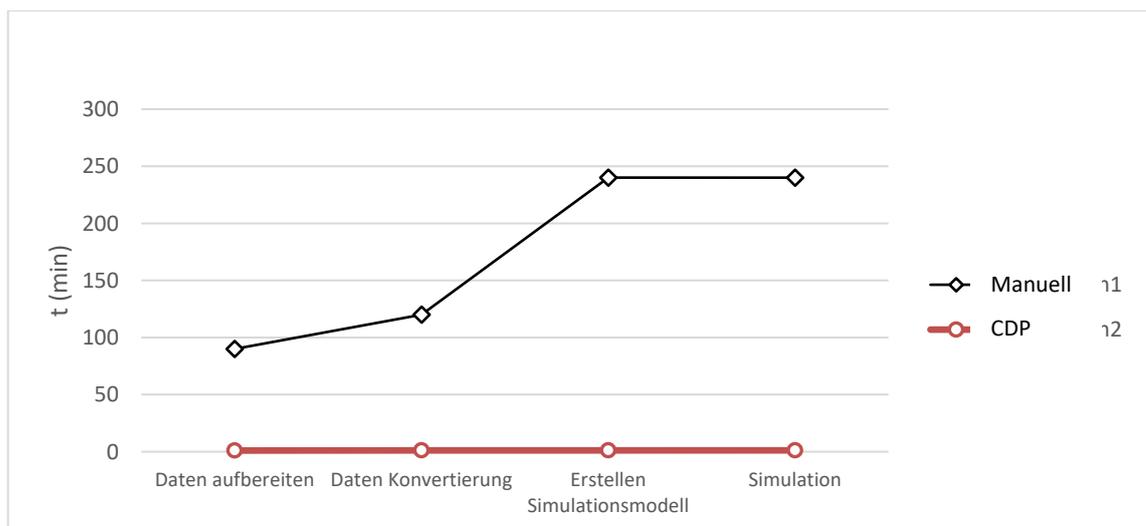


Abbildung 26 Dauer des Entwurfsprozesses (Quelle: Lehrstuhl für Architekturinformatik)

Die Ursache für die zeitliche Diskrepanz zwischen den Bearbeitungsschritten liegt am mehrmaligen Wechsel der Zeichenprogramme im konventionellen Prozess (Abbildung 26). Daten müssen nach jedem Schritt lokal gespeichert, gefiltert und abgerufen werden. Eine Aufbereitung und Kontrolle der Zeichnungsdaten führte zu Verzögerungen im Entwurfsprozess. Zudem steigt der Mehraufwand bei nachträglichen Entwurfsänderungen. Der Aufbau des Simulationsmodells ist ein weiterer Faktor für den zeitlichen Versatz. Dabei werden jedem Element manuell numerische Werte zugewiesen und mögliche Pfade für die Agenten erstellt. Die Simulationsdauer ist bei beiden Prozessen identisch, da bei beiden Methoden auf den Berechnungskern von *crowd:it* zugegriffen wird.

6.5 Qualitativer Vergleich

Dem quantitativen Vergleich folgt ein qualitativer Vergleich, dabei werden die Vor- und Nachteile der einzelnen Prozessschritte einem konventionellen- und automatisierten Prozess gegenübergestellt werden (Tabelle 13).

Tabelle 13 Qualitative Bewertung der Prozesse

Qualitative Bewertung	Konventioneller Prozess	Automatisierter Prozess (CDP)
Datenimport		
Planungsgebiet wählen	+	+
Datenaustausch zum Nachbearbeiten	-	+
Nachbearbeitung		
Maßstab Überprüfung	-	+
Nachbearbeiten notwendig	-	+
Datenaustausch zum Entwerfen	-	+
Entwurfsphase		
Raumprogramm erstellen	0	0
Interaktiver Entwurfsprozess	-	+
Datenaustausch zur Simulation	-	+
Simulation		
Simulationsmodell festlegen	-	+
Simulationssettings Standardwerte	+	+
Simulationssettings modifizierbar	+	-
Simulationsdauer	0	0
Kollaborativer Prozess	-	+

Vorteile (+)

Mithilfe von *OpenStreetMap*-Daten kann das Planungsgebiet bei beiden Varianten einfach und schnell mit korrekten Geoinformationsdaten bestimmt und geladen werden. Beim manuellen Prozess ergeben sich Vorteile beim Anpassen der Simulations-Voreinstellungen. Die Vorteile des automatisierten Prozesses überwiegen, aufgrund des fehlerfreien Datenaustauschs in wenigen Sekunden. Ein nachträgliches Bearbeiten der Geometriedaten des Königsplatzes war nicht notwendig. Die Möglichkeit, Gebäude zu löschen oder zu verändern ist mithilfe der interaktiven Eingabemöglichkeiten der *CDP* vorhanden. Neben der schnellen Verarbeitung von Daten im automatisierten Prozess, fordern weitere Funktionen wie eine interaktive Entwurfsumgebung den Entwurfsprozess. Ein kollaborativer Ansatz der *CDP* soll den Planenden bei der Entscheidungsfindung unterstützen und zusätzlich zu besseren Ergebnissen in frühen Entwurfsphasen führen.

Neutral (o)

Beim Erstellen des Raumprogramms und der Simulationsdauer sind weder Vor- oder Nachteile beider Prozessvarianten vorhanden.

Nachteile (-)

Eines der größten Nachteile beim konventionellen Prozess ist der Zeitaufwand beim Datenaustausch zwischen den Programmen und die Kontrolle eines korrekten Datenimports. Gedanklich muss sich der Anwender auf die neue verschiedenen Benutzeroberflächen einstellen. Durch die Unterbrechungen eines Softwarewechsels, wird der Denkprozess eines Anwenders partiell unterbrochen was zusätzlich zu einer Verzögerung beiträgt. Für die Simulation sind Voreinstellungen in *crowd:it* hinterlegt. Die Voreinstellungen für die Simulation betreffen Durchschnittswerte für die Auflösung der Gitterstruktur im Planungsgebiet, der Agentengeometrie und dem Agentenverhalten.

Beim automatisierten Prozess in *CDP* müssen diese Werte direkt im Programmcode verändert werden. Für eine schnelle Abschätzung im Maßstab 1:500 sind die voreingestellten Werte zulässig und eine Anpassung der Standardwerte ist nur im Ausnahmefall notwendig. Die Automatische Datenübernahme der *CDP* aus *OSM* hat einen weiteren Nachteil: Die *OSM* Bestandsdaten können aktuell ausschließlich online bearbeitet werden und Änderungswünsche müssen durch die Community bestätigt werden. Durch die Eingabemöglichkeiten in der *CDP* können bereits geladene Bestandsdaten lokal angepasst und gespeichert werden, was diesen Nachteil relativiert. Ein weiterer Nachteil besteht, durch spezifische Anpassungswünsche der Simulationsparameter durch Akteure. Eine solche Änderung von Parametern muss beim automatisierten Prozess durch Entwickler im Programmcode überarbeitet werden.

Zusammengefasst überwiegen die Vorteile des automatisierten Prozesses der *CDP*. Durch den automatisierten Prozess werden Fehler eines Datenaustauschs vermieden und der Zeitaufwand im Entwurfsprozess erheblich reduziert.

7 Zusammenfassung und Ausblick

Ziel dieser Arbeit war es, eine interaktive Entwurfsplanung unter Berücksichtigung von Personenströmen aufzuzeigen, wodurch Ressourcen und Kosten in der Planung von Gebäuden, öffentlichen Plätzen und Veranstaltungen eingespart werden können.

In der Praxis werden Personenstromsimulationen aufgrund von zeitlichem Aufwand und Komplexität teilweise vollständig vernachlässigt. Stattdessen ist ein normativer Ansatz zur Berechnung von Kapazitätsgrenzen üblich, um Einschätzungen zur Sicherheit und dem vorkehrenden Brandschutz zu treffen. Für Sonderbauten und (Groß-)Veranstaltungen können Personenstromsimulationen zu mehr Sicherheit in der Planung beitragen. Aufbauend auf den Forschungsfragen und konkreten Anwendungsszenarien konnten neue Interaktionsmethoden (**Eingabe**), Techniken zum automatisierten Aufbau von Simulationsmodellen und zur Ansteuerung der Simulation (**Simulation**) und neue Formen der Präsentation von Simulationsergebnissen (**Visualisierung**) gefunden werden:

Eingabe

Vereinfachte und vorweggenommene Inputparameter verhelfen der Nutzer*in zu einem schnellen Einstieg im Umgang mit Personenstromsimulationen auf der *CDP*. Je nach Anforderung an die Entwurfsaufgabe und an das Untersuchungsgebiet können die Simulationsmodelle und -parameter im Entwurfsprozess angepasst werden. In dieser Arbeit wurde eine zeiteffiziente Methode zur Berechnung und Darstellung von Personenströmen vorgestellt. In Zukunft kann erforscht werden, wie sich die Veränderung von einzelnen Parametern auf die Genauigkeit von Simulationsmodellen für Personenströmen auswirkt.

Simulation

Die Forschungsfragen der Simulation konzentrierten sich im Wesentlichen auf die Verkürzung der Simulationszeit und der Kommunikation von Zwischenergebnissen zur Entwurfsunterstützung. Durch die Überlagerung von Simulationsergebnissen, die Anpassung von Parametern und der Schaffung einer Schnittstelle konnten die gestellten Anforderungen erfüllt werden. Dabei sollen die Simulationsergebnisse in Echtzeit ein erstes Feedback zu Kapazitätsengpässen geben. Im weiteren Planungsverlauf müssen die Ergebnisse von Experten geprüft und validiert werden. Eine Anpassung von Back-End-Parametern konnte die Simulationszeit reduzieren. Der Informationsaustausch zwischen *crowd:it* und der digitalen Entwurfsplattform *CDP* wurde über eine performante Schnittstelle hergestellt, somit können Zwischenergebnisse kommuniziert und hohe Rechenkapazitäten ausgelagert werden. Zudem können (CAAD-)Softwareentwickler*innen die Open-Source-Schnittstelle nutzen, um *crowd:it* an weitere

Entwurfswerkzeuge anzubinden. Damit ist es möglich, den frühen Entwurf mit Personenstromsimulationen in Echtzeit zu evaluieren.

Visualisierung

Mit der Visualisierung durch eine *Heatcloud* konnten die Simulationoutputs in einem angemessenen Detailgrad dargestellt werden. Die Überlagerung und Kombination von Informationen stellt ein entscheidendes Element zur Bewertung von Entwurfsvarianten dar. Das Aufzeigen neuer Visualisierungsstrategien, war ein Hauptbestandteil dieser Forschungsarbeit. Die Nutzerfreundlichkeit und Lesbarkeit von Animationen bzw. Diagrammen konnten mithilfe von Selbsttests bewertet werden. Aufgrund der COVID-Pandemie war es bedauerlicherweise nicht möglich, umfassende Nutzerstudien durchzuführen. Für eine wissenschaftliche Bewertung der Bedienbarkeit und Verständlichkeit von Visualisierungsmethoden, bedarf es einer genaueren Untersuchung durch externe Teilnehmende. Darüber hinaus wurde in Kapitel 4.4 das Konzept einer erweiterten Displayanzeige zur Darstellung zusätzlicher Informationen vorgestellt.

Mit dieser Forschungsarbeit wurde die Grundlage geschaffen, um Entwurfsentscheidungen mithilfe von Personenstromsimulationen und Anforderungen an den räumlichen Komfort in Echtzeit zu optimieren. Dies geschah auf Grundlage von validen Simulationsergebnissen in Anbindung an das Softwareprodukt *crowd:it*. Dieses Forschungsprojekt zeigt auf, wie Planungsfehler in Hinblick auf Kapazitätsgrenzen, Fluchtwegführung und Fluchtwegbreiten im frühen Entwurfsprozess erkannt und vermieden werden können. Dadurch reduziert sich der Planungsaufwand in späteren Entwurfsphasen. Das Werkzeug trägt somit zu Planungs- und Kostensicherheit von Bauprojekten bei. Im Vergleich zum konventionellen Prozess und der Beauftragung externer Simulationsspezialisten, ist die vorgestellte Methode zeiteffizient und beliebig oft wiederholbar, während man durch externe Planungsbüros simulierte Ergebnisse oft mehrere Stunden bzw. Tage warten muss. Im Fokus dieser Arbeit stand eine kollaborative, interaktive Eingabe durch die Nutzer*in. Eine haptische, einfach bedienbare Benutzeroberfläche gekoppelt mit Simulationsergebnissen in Echtzeit ermöglicht es mehreren Nutzer*innen, gleichzeitig am Entwurfsprozess zu partizipieren. So wird der Designprozess zum Verhandlungsprozess aller Akteure auf gleicher Augenhöhe. Die Auswirkungen von Entwurfsanpassungen werden unmittelbar aufgezeigt und beeinflussen somit die Diskussion und Entwurfsentscheidungen.

A1 Abkürzungsverzeichnis

API	Application Programming Interface
CAAD	Computer-aided architectural design
CDP	Collaborative Design Platform
C#	C-Sharp, objektorientierte Programmiersprache
C++	Programmiersprache für Systeme und Anwendungen
DXF	Drawing Interchange Format
Fps	Frames per second
Java	Objektorientierte Programmiersprache
OSM	Open Street Maps
RGB	R = rot, G = grün, B = blau
SDK	Software Development Kit
UI	User Interface

A2 Abbildungen

<i>Abbildung 1 Prototypische Umsetzung</i>	8
<i>Abbildung 2 CDP // Collaborative Design Platform – etablierte Entwurfswerkzeuge und digitale Tools nahtlos verbunden</i>	14
<i>Abbildung 3 crowd:it – Simulationsmodell (Quelle: accu:rate)</i>	15
<i>Abbildung 4 Szenario „Indoor“ - Hochhaus</i>	19
<i>Abbildung 5 Szenario „Outdoor“ - Königsplatz</i>	21
<i>Abbildung 6 Parametereingabe durch intuitive Interaktionen mit physischen Objekten</i>	23
<i>Abbildung 7 Finites vs. kontinuierliches Simulationsmodell</i>	30
<i>Abbildung 8 Darstellung von Simulationsergebnissen direkt im Entwurfskontext</i>	33
<i>Abbildung 9 Agentenpositionen (in 3 Schritten)</i>	36
<i>Abbildung 10 Agenten-Pfade</i>	37
<i>Abbildung 11 „Occupancy“-Heatmap und „Significant Congestion“-Heatmap</i>	38
<i>Abbildung 12 Kombination der Darstellungstechniken</i>	39
<i>Abbildung 13 Mockup der Heatcloud</i>	40
<i>Abbildung 14 Progressive Berechnung der gemittelten Agentendichte über den gesamten Simulationszeitraum</i>	42
<i>Abbildung 15 Simulationsergebnisse</i>	43
<i>Abbildung 16 Annäherung durch vorläufige Ergebnisse</i>	44
<i>Abbildung 17 Prototyp des Informationsbildschirms</i>	45
<i>Abbildung 18 Informationsdisplay als Ergänzung der horizontalen Anzeige</i>	46
<i>Abbildung 19 Sequence Diagram</i>	47
<i>Abbildung 20 Kommunikationsprotokoll</i>	50
<i>Abbildung 21 Königsplatz, OSM Daten Export</i>	58
<i>Abbildung 22 Elk-Plugin Schaltung (Rhinoceros 6, Grasshopper 3D)</i>	58
<i>Abbildung 23 Ergänzen von fehlenden Gebäuden mithilfe von ArchiCAD</i>	59
<i>Abbildung 24 Entwurf eines Festivalgeländes am Königsplatz</i>	61
<i>Abbildung 25 Aufbau eines Simulationsmodells in crowd:it</i>	63
<i>Abbildung 26 Dauer des Entwurfsprozesses</i>	65

Abbildungshinweis:

Alle Abbildungen wurden selbst erstellt (Lehrstuhl für Architekturinformatik), sofern keine Quellen im Abbildungsverzeichnis angegeben wurden.

A3 Tabellenverzeichnis

<i>Tabelle 1 Agenten Einstellungen</i>	25
<i>Tabelle 2 Einstellung vom Agentenverhalten</i>	25
<i>Tabelle 3 Einstellungen zum Simulationsgebiet</i>	25
<i>Tabelle 4 Entwurfsobjekte im Kontext der Simulationsszenarien</i>	27
<i>Tabelle 5 Veränderung von Quellen</i>	31
<i>Tabelle 6 Veränderung von Wartezonen</i>	32
<i>Tabelle 7 Veränderung von Zielen</i>	32
<i>Tabelle 8 Standort München</i>	57
<i>Tabelle 9 Funktionsprogramm</i>	60
<i>Tabelle 10 Agenten Einstellungen</i>	61
<i>Tabelle 11 Einstellung zum Agentenverhalten</i>	62
<i>Tabelle 12 Einstellungen zum Untersuchungsgebiet</i>	62
<i>Tabelle 13 Qualitative Bewertung der Prozesse</i>	66

A4 Quellen- und Literaturverzeichnis

- accu:rate (2021): Unsere Software crowd:it. Online verfügbar unter <https://www.accu-rate.de/de/software-crowd-it-de/>, zuletzt aktualisiert am 11.02.2021, zuletzt geprüft am 25.02.2021.
- Akgul, Faruk (2013): ZeroMQ. Online-Ausg. Birmingham: Packt. Online verfügbar unter <https://ebookcentral.proquest.com/lib/kxp/detail.action?docID=1142880>.
- Attia, Shady; Gratia, Elisabeth; Herde, André de; Hensen, Jan L.M. (2012): Simulation-based decision support tool for early stages of zero-energy building design. In: *Energy and Buildings* (49), S. 2–15.
- Bengel, Günther (2014): Client-Server Architecture. Berlin/Heidelberg: Springer-Verlag.
- Bernard, Trevor (2021): zeromq/jeromq. Online verfügbar unter <https://github.com/zeromq/jeromq>, zuletzt aktualisiert am 19.02.2021, zuletzt geprüft am 19.02.2021.
- Bitzer, Timo Florian (2010): Mikroskopische Modellierung und Simulation des Fußgängerverkehrs. Dissertation. Universität Stuttgart, Stuttgart.
- Bratoev, Ivan; Schubert, Gerhard; Chokhachian, Ata; Petzold, Frank; Auer, Thomas (2018): Interactive Solar Potential Simulation for Early Stage Urban Planning. In: BAUSIM 2018. Karlsruhe, Germany, S. 407–411.
- Comer, Douglas E.; Stevens, David L. (1996): Client-server programming and applications. BSD socket version. 2. ed.
- Dietrich, Felix; Köster, Gerta; Seitz, Michael; Sivers, Isabella von (2014): Bridging the gap: From cellular automata to differential equation models for pedestrian dynamics. In: *Journal of Computational Science* 5 (5), S. 841–846. DOI: 10.1016/j.jocs.2014.06.005.
- Holl, Stefan (2016): Methoden für die Bemessung der Leistungsfähigkeit multidirektional genutzter Fußverkehrsanlagen. Jülich: Forschungszentrum Jülich GmbH.
- Kneidl, Angelika (2013): Methoden zur Abbildung menschlichen Navigationsverhaltens bei der Modellierung von Fußgängerströmen. Dissertation. Technische Universität München, München.
- Kretz, Tobias: Pedestrian Traffic. Simulation and Experiments. Dissertation. Universität Duisburg-Essen, Essen. Online verfügbar unter https://www.researchgate.net/publication/29799949_Pedestrian_Traffic_-_Simulation_and_Experiments, zuletzt geprüft am 20.03.2020.
- Petersen, Steffen; Svendsen, Svend (2010): Method and simulation program informed decisions in the early stages of building design. In: *Energy and Buildings* (42), S. 1113–1119.
- Petzold, Frank; Auer, Thomas (2018): Simulationsgestützte Entwurfsplanung im städtebaulichen Kontext unter Berücksichtigung energetischer und raumklimatischer Aspekte. Stuttgart: Fraunhofer IRB Verlag (Forschungsinitiative Zukunft Bau, F 3100).
- Rittel, Horst W.J.; Webber, Melvin M. (1973): Dilemmas in a General Theory of Planning. In: *Policy Sciences* 4, S. 155–169.
- Ritter, Fabian; Schubert, Gerhard (2014): Integration digitaler Gebäudesimulationen in den frühen Entwurfsphasen. In: Proc. of the 26th Forum Bauinformatik, S. 249–258.

- Roetzel, Astrid (2015): Occupant behaviour simulation for cellular offices in early design stages. Architectural and modelling considerations. In: *Building Simulation* (8), S. 211–224.
- Schubert, Gerhard (2014): Interaktionsformen für das digitale Entwerfen. Dissertation. Technische Universität München, München.
- Schubert, Gerhard; Bratoev, Ivan; Strelchenko, Vadym; Petzold, Frank (2019): I Hear, What You Are Doing! Workspace Awareness in Collaborative Virtual Environments. In: *Intelligent & Informed - Proceedings of the 24th CAADRIA Conference*, Bd. 1, S. 583–592.
- Schubert, Gerhard; Strobel, Benjamin; Petzold, Frank (2016): Tangible Mixed Reality. Interactive augmented visualisation of digital simulation in physical working models. In: S. Chien, S. Choo, M. A. Schnabel, W. Nakapan, M. J. Kim und S. Roudavski (Hg.): *Living Systems and Micro-Utopias. Towards Continuous Designing*, Proceedings of the 21st International Conference of the Association for Computer-Aided Architectural Design Research in Asia CAADRIA, S. 333–342.
- Seitz, Michael J.; Köster, Gerta (2012): Natural discretization of pedestrian movement in continuous space. In: *Physical review. E, Statistical, nonlinear, and soft matter physics* 86 (4). DOI: 10.1103/PhysRevE.86.046108.
- Serginson, Michael; Mokhtar, George; Kelly, Graham (2013): A Theoretical Comparison of Traditional and Integrated Project Delivery Design Processes on International BIM Competitions. In: *International Journal of 3-D Information Modeling* 2 (4), S. 52–64.
- Sivers, Isabella von; Köster, Gerta (2015): Realistic Stride Length Adaptation in the Optimal Steps Model. In: Mohcine Chraïbi, Maik Boltes, Andreas Schadschneider und Armin Seyfried (Hg.): *Traffic and Granular Flow '13*. Cham. Conference on Traffic and Granular Flow. Cham: Springer, S. 171–178.
- Sivers, Isabella von; Templeton, Anne; Köster, Gerta; Drury, John; Philippides, Andrew (2014): Humans do not Always Act Selfishly: Social Identity and Helping in Emergency Evacuation Simulation. In: *Transportation Research Procedia* 2, S. 585–593. DOI: 10.1016/j.trpro.2014.09.099.
- The Qt Company (2021): Qt | Cross-platform software development for embedded & desktop. Online verfügbar unter <https://www.qt.io/>, zuletzt aktualisiert am 24.02.2021, zuletzt geprüft am 25.02.2021.