

Geosensornetze

WS 2013/2014

Hausarbeit von
Andre Lehnert und Marcell Salvage

27. Januar 2014

Inhaltsverzeichnis

1	Einführung	1
1.1	Aufgabenbeschreibung	1
2	Simulationsumgebung	3
2.1	Benutzerschnittstelle	3
2.2	Personen	3
2.2.1	Zustände	3
2.2.2	Bewegungsmodell	3
2.2.3	Kommunikationsmodell	3
2.3	Notausgänge	5
2.3.1	Zustände	5
2.4	Gefahrensituationen	5
2.4.1	Zustände	5
2.5	Patches	5
2.5.1	Implizite Zustände	5
2.6	Ressourcen der Simulationsumgebung	5
3	Algorithmik	10
4	Evaluation	12
4.1	Effizienz	12
4.2	Fazit	12
	Literaturverzeichnis	i

1 Einführung

Mit Hilfe der NetLogo-Simulationsumgebung [3] wird eine dynamische Evakuierung von Gebäuden implementiert. Dazu werden die Grundrisse der Gebäude oder Etage in die Simulationsumgebung geladen. Diese dient als Grundlage für die Platzierung von Personen, Gefahrenereignissen und Notausgängen.

Auf der Flucht vor Gefahrenereignissen werden die Personen von mobilen Geräten unterstützt, die zur Warnung anderer Personen und zur Lokalisierung der Notausgänge dienen.

1.1 Aufgabenbeschreibung

Die Aufgabe besteht in der Umsetzung einer geeigneten Simulationsumgebung (siehe Kapitel 2). Auf deren Basis Algorithmen zur Lokalisierung und Bestimmung eines Fluchtweges zu den Notausgängen entwickelt werden (siehe Kapitel 3). Schließlich wird eine Evaluation der Algorithmen in Punkto Effizienz und Zuverlässigkeit durchgeführt und Reflektiert (siehe Kapitel 4).

Personen werden in der NetLogo-Umgebung als Agenten realisiert, die sich nach dem Bewegungsmodell (siehe Abschnitt 2.2.2) innerhalb des Grundrisses bewegen. Die initiale Platzierung geschieht zufällig, analog zu der Platzierung der Gefahrenereignisse. Personen besitzen die Fähigkeit diese Gefahrenereignisse in ihrer Umgebung wahrzunehmen und als Gefahrensituation zu deuten. Die Personen versuchen daraufhin den besten Weg zu einem Notausgang zu finden und benachbarte Personen dabei über ihre mobilen Geräte zu warnen.

Als Gefahrensituationen (siehe Abschnitt 2.4) zählt eine gewisse Anzahl von Giftgasbomben mit eingebautem Zeitzünder, die je ein Gefahrenereignis darstellen. Die freigesetzten Gasmengen sind regulierbar und breiten sich innerhalb des freien Raumes aus.

Zur Evakuierung der Personen aus dem Gefahrenereignis werden Notausgänge (siehe Abschnitt 2.3) in dem Grundriss platziert, deren Position sich während der Simulation nicht ändert, sogenannte *anchor notes*.

Eine feste Position ist notwendig zur Realisierung der dezentralen Lokalisierungs-

algorithmen, die auf den mobilen Geräten der Personen aktiv sind und bei einer dynamischen Evakuierung assistieren. Durch eine lokal eingeschränkte Kommunikationsfähigkeit (siehe Abschnitt 2.2.3) werden Informationen über die Passierbarkeit der Notausgänge an die mobilen Geräte verteilt. Dies ermöglicht die sichere Evakuierung, falls beispielsweise das Giftgas einen Notausgang erreicht hat oder die Fluchtwege blockiert sind.

2 Simulationsumgebung

Das Kapitel der Simulationsumgebung befasst sich mit den Eingabemöglichkeiten zur Anpassung der Simulationsparameter, sowie der konkreten Umsetzung der simulierten Welt, mit der Logik für die Notausgänge, der Personen, der Gefahrensituationen und dem Kommunikationsmodell.

Die Ausgangsbasis für die verwendeten Modelle und Protokolle stammen aus den erstellten Übungen zur Vorlesung Gensensornetze, sie wurden jedoch auf die Anforderungen der Hausarbeit adaptiert.

2.1 Benutzerschnittstelle

2.2 Personen

2.2.1 Zustände

2.2.2 Bewegungsmodell

Dieser Abschnitt beschreibt zunächst das Bewegungsmodell der Personen ohne aktive Gefahrensituation.

Algorithmus 1 random-walk

```
nb ← one-of neighbors
while [patch-state] of nb = WALL do
  nb ← one-of neighbors
end while
face nb
forward 1
```

2.2.3 Kommunikationsmodell

Protokoll 2 Warnung vor Gefahrensituationen

State Trans. Sys.: $\langle \{\text{INIT, EVENT_DETECTED, FLEEING, RESCUED, DEAD}\} \rangle$

Initialization: All nodes in state INIT

Restrictions: Reliable communication; connected, bidirected communication graph $G = (V, E)$, neighborhoodfunction nbr: $V \rightarrow 2^V$

Local data:

INIT

Receiving(*event_detected*)

while *not event_detected* **do**

 random-walk

 create-graph(G)

; generate complete new graph

end while

become EVENT_DETECTED

EVENT_DETECTED

broadcast(*event_detected*)

; broadcast event detection to linked neighbors

become FLEEING

FLEEING

broadcast(*event_detected*)

; rebroadcast event detection to linked neighbors

while *not person-reach-exit* **do**

 person-move-to-exit

; using orientation-algorithm

 create-graph(G)

; generate complete new graph

if touching-gas **then**

 become DEAD

end if

end while

become RESCUED

RESCUED

create-graph(G)

; generate complete new graph without note

DEAD

create-graph(G)

; generate complete new graph without note

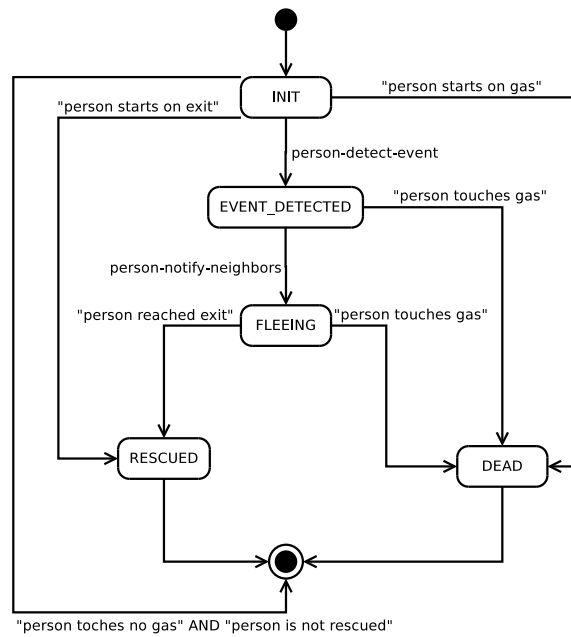


Abbildung 2.1: Zustandsdiagramm der Personen

2.3 Notausgänge

2.3.1 Zustände

2.4 Gefahrensituationen

2.4.1 Zustände

2.5 Patches

2.5.1 Implizite Zustände

2.6 Ressourcen der Simulationsumgebung

Die Simulationsumgebung wird über die Datei `Evakuierung.nlogo` gestartet. Der Programmcode ist nach Funktion und *Breed*-Klasse unterteilt.

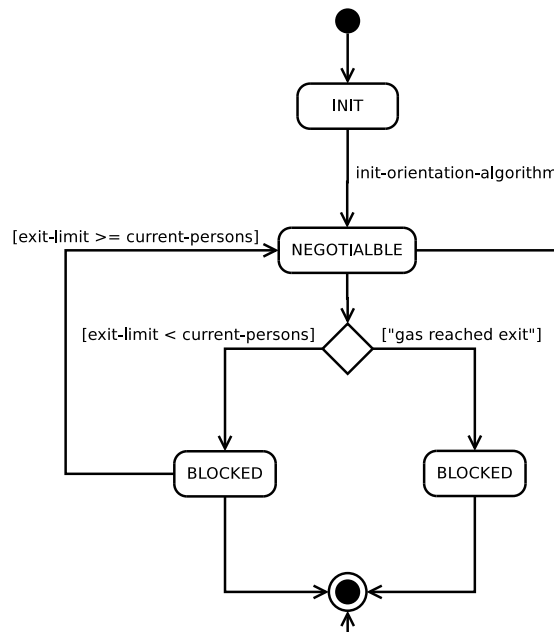


Abbildung 2.2: Zustandsdiagramm der Notausgänge

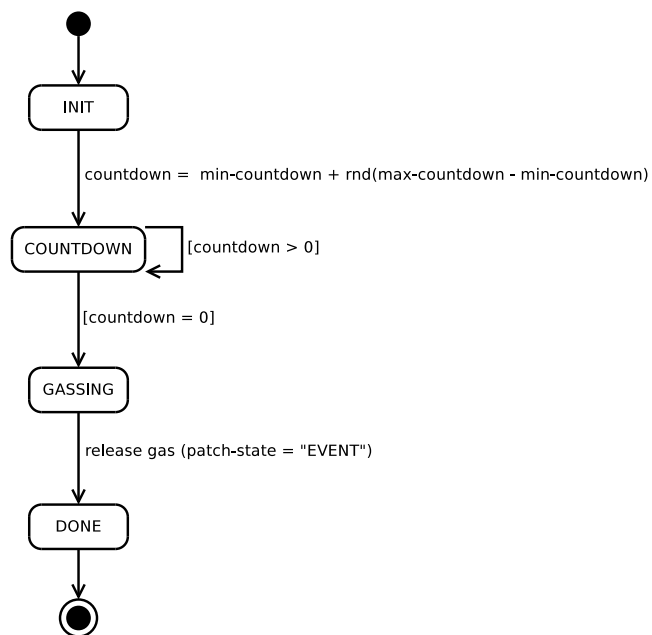


Abbildung 2.3: Zustandsdiagramm der Gefahrenereignisse

Protokoll 3 Gefahrensituation

State Trans. Sys.: $\langle \{\text{INIT}, \text{COUNTDOWN}, \text{GASSING}, \text{DONE}\} \rangle$

Initialization: All nodes in state INIT

Restrictions: All patches in state $\{\text{NONE}, \text{DONE}\}$

Local data: countdown $\in \mathbb{N}_{\geq 0}$

INIT

Spontaneously

countdown \leftarrow minCountdown + (random (maxCountdown - minCountdown))

become COUNTDOWN

COUNTDOWN

countdown \leftarrow countdown - 1

if countdown = 0 **then**

 become GASSING

end if

GASSING

ask patch-here [patch-state \leftarrow EVENT]

become DONE

DONE

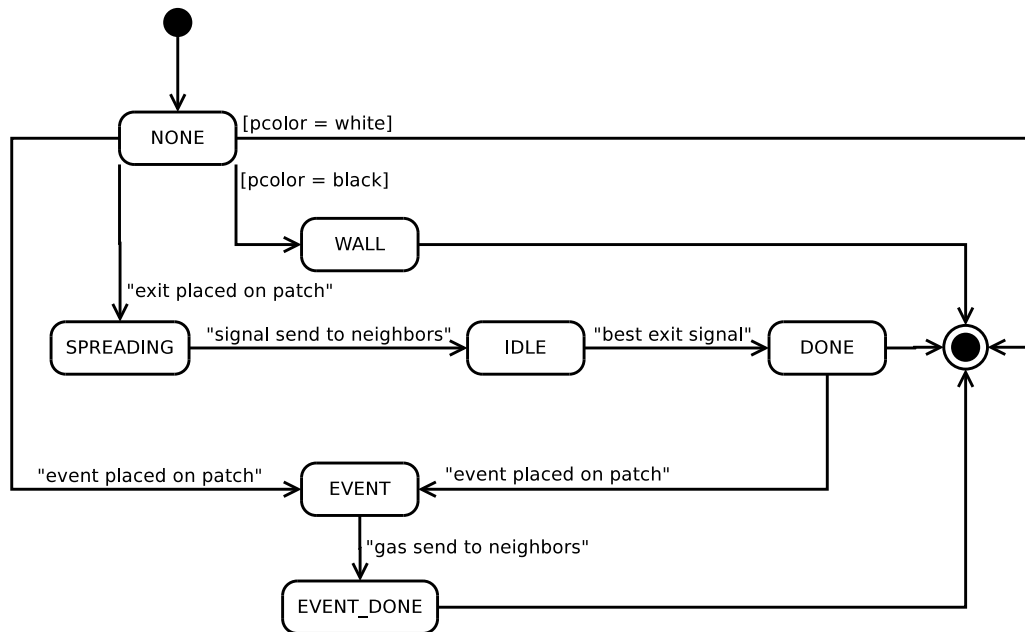


Abbildung 2.4: Zustandsdiagramm der Patches

Gefahrensituationen

event.nls
event-gassing.nls

Die Quellcode-Datei `event.nls` beinhaltet den Lebenszyklus der Gefahrenereignisse und deren Zustandsübergangsprotokoll. Mittels `event-gassing.nls` wird die Ausbreitung des Giftgases implementiert, die größtenteils den Patch-Lebenszyklus manipuliert.

Lokalisierung

locate.nls

In dieser Datei wird der Algorithmus zur Lokalisierung aus dem Paper [2] implementiert.

Notausgänge

```
exit.nls  
exit-cellular-automaton.nls  
exit-gsn.nls
```

Die Quellcode-Datei `event.nls` steuert den Setup und den Lebenszyklus der Notausgänge. Mittels `exit-cellular-automaton.nls` wird der Orientierungsalgorithmus auf Basis des zellulären Automats realisiert. Letztlich definiert die Datei `exit-gsn.nls` das Kommunikationsmodell zur Übermittlung der Statusinformationen.

Personen

```
person.nls  
person-gsn.nls  
person-linking.nls
```

`person.nls` regelt den Setup der Personen und deren Lebenszyklus. `person-gsn.nls` umfasst den Quellcode für die Kommunikation zwischen Personen und mit der Datei `person-linking.nls` wird der Graph zwischen Personen erstellt.

Simulationswelt

```
patch.nls
```

Hier wird der Quellcode für den Lebenszyklus der *Patches* definiert.

3 Algorithmik

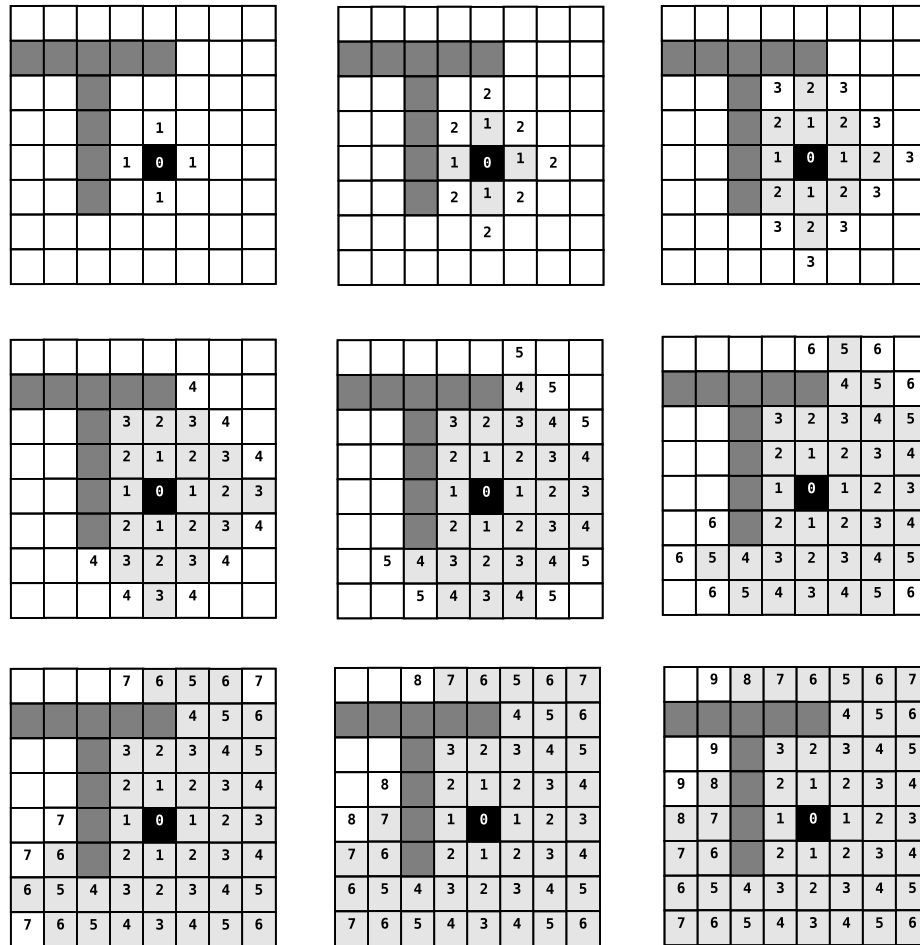


Abbildung 3.1: Fluten der Patches (Zellulärer Automat)

4 Evaluation

4.1 Effizienz

4.2 Fazit

Literaturverzeichnis

- [1] Isaac Amundson and Xenofon D. Koutsoukos. *A Survey on Localization for Mobile Wireless Sensor Networks*. Department of Electrical Engineering and Computer Science, Vanderbilt University.
- [2] Jonathan Bachrach, Radhika Nagpal, Michael Salib and Howard Shrobe. *Experimental Results for and Theoretical Analysis of a Self-Organizing Global Coordinate System for Ad Hoc Sensor Networks*. Telecommunication Systems, page 213–233. 2004.
- [3] Uri Wilensky. *Netlogo*. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. 1999. <http://ccl.northwestern.edu/netlogo/>, Stand: 26.01.2014.