Modeling pedestrians in an airport scenario with a time-augmented Petri net

Smits, D.; Visser, A.; Groen, F.

DOI
10.1007/978-3-642-33932-5_50

Publication date
2013

Document Version
Submitted manuscript

Published in
Advances in intelligent systems and computing

Modeling pedestrians in an airport scenario with a time-augmented Petri net

Djura Smits Arnoud Visser Frans Groen
Intelligent Systems Lab Amsterdam, Universiteit van Amsterdam
P.O. Box 94323, 1090 GH Amsterdam, The Netherlands

***Abstract*—In this paper time-augmented Petri nets are used to model people in the transit hall of an airport. Their behavior is strongly influenced by an event with a clear deadline (their flight), but typically there is so much time left that they linger and can be tempted to show random other behaviors, often induced by the location (encountering a coffee corner or a toilet). All behaviors are stochastic, but the firing rate is made a function of both location and time. This framework allows to show a rich set of behaviors; the diversity of the emergent behaviors is initiated with probabilities from observations in an actual transit hall of an airport.**

I. Introduction

Simulation of pedestrians is a widely studied topic. Simulation could predict the emergent patterns in the movements of people, for instance when monitoring them in public places [1]. The local interactions of multiple people generate regular patterns of motion, although less frequent interactions could be interpreted as suspicious behavior. The detection of this suspicious behavior would be the reason why the pedestrians are monitored in the first place, yet it could be easily confused with less frequent but normal behavior as for instance a patrolling police officer or somebody waiting on another. By enriching the behaviors of pedestrians, the addition of suspicious behavior can be made less obvious, which allows to easily create artificial testing data for suspicious behavior detection algorithms and such. In this paper the enrichment of the behaviors is made possible by initiating the interaction of the environment.

Another way to look at the behavior of groups of pedestrians is to explicitly specify how they behave when they enter a certain type of situation [3]. In our approach, we would like to be able to specify behavior by drawing situations on an environment. For example, when a food stand is present in the area, we would like to be able to indicate that the agents that are placed in the neighborhood of that stand are able to perform a certain food-buying behavior, and by doing so, inject this behavior into the pedestrians nearby.

An additional requirement is that we would like to indicate a certain deadline for an agent which is the time at which its goal is not reachable any more. This approach results in roughly two types of behavior, hurried and relaxed. When the deadline approaches, less and less actions are likely to be done, since actions that take much time would result in not reaching the goal in time.

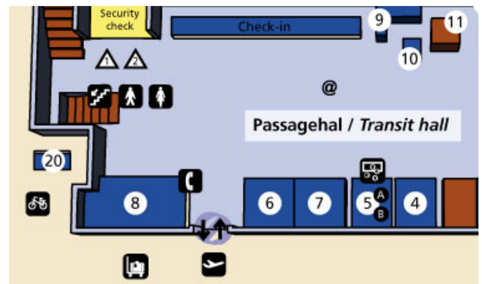

Fig. 1. A snapshot of the map of Rotterdam Airport. The numbers on the map indicate a luggage locker (9), meeting point (10), elevator (11), smoking shelter (20), shops (4,5,7) and information desks (6,8). Courtesy Rotterdam The Hague Airport.

We validated our model by mimicking certain behaviors found on Rotterdam airport. We chose Rotterdam airport because time restrictions are very prevalent in a transit hall like the one on Rotterdam airport (see Fig. 1).

II. Related Work

As previously mentioned, most studies focus on aggregated crowd behavior or the behavior of an individual in such crowd [3]. In such cases the interaction with the environment is mainly reduced to collision detection. This greatly limits the applicability of the simulation. In many situations there is not a crowd with many people doing the same, but a situation where individuals could be distinguished, each individual doing something different. The obvious solution is to extend the knowledge of the pedestrians with instructions about how to interact with each other or with objects in the environment. This has been successfully done for instance by Shao and Terzopoulos [4]. They used an extensive psychological model to determine the behavior of the individuals. This led to a simulation in which the behavior looks very realistic, even when one person is followed for a long time. The downside to this method is that the behavioral model for the pedestrians has to be specifically crafted for the environment, which will be very time consuming. It would be easier to generate the virtual human agents if the environment would automatically decide for the agents what interactions are possible and appropriate. The first step towards this focus was made by Kallmann and Thalmann who introduced the principle of smart objects [5].

Their agents' behavior is determined by both their current condition (e.g. hungry or tired) and the available objects placed in the environment. In the most basic approach, these smart objects take complete control of the agents for a short period of time and let them interact with the object. The big advantage of this method is that the information for interacting with a specific object does not need to be stored in the pedestrians, but is stored in the objects themselves. This way, additional behavior can be added to a pedestrian by simply placing a new object in the environment. The object also keeps track about how many agents can interact with it at the same time and if the interaction should be the same for all agents. For instance, an elevator modeled as a smart object will make the first agent interacting with it press the button, but not the next agents that approach this object. By using smart objects, the internal model of the pedestrians can be kept very simple, because they do not need to remember specific information about how to interact with the objects. Furthermore, this means that the pedestrians do not have to be specifically designed for the current simulation environment, because the environment will tell them how to act. In the most basic approach to smart objects, the agents lose all their autonomy when they approach a smart object.

The challenge approached in this paper is to combine the smart object approach with time-driven behaviors; the pedestrians are typically on a location with a goal and have to perform a number of behaviors before a deadline. To model the time-aspect we have chosen to use Petri nets, as described in the next section.

III. Method

Petri nets are a mathematical modeling language used for the description of distributed systems [7]. A Petri net is a bipartite graph consisting of two types of nodes: places and transitions. These nodes are connected by directed arcs. An arc can run from either a place to a transition, or from a transition node to a place, but never from a place to a place, or between two transitions. Activity in a Petri net is expressed by the movement of tokens from place to place, through transitions. Input arcs (from place to transition) denote which places need to contain tokens in order to enable the transition. When a transition is enabled, it consumes the tokens from the input places, and produces tokens in the place indicated by the output arc. Basic Petri nets can be described by a five-tuple:

$$PN = (P, T, I, O, M_0) \tag{1}$$

which comprises of

- a set of places $P = (p_1, p_2, ..., p_m)$,
- a set of transitions $T = (t_1, t_2, ..., o_m)$,
- a set of input arcs $I \subset P \times T$,
- a set of output arcs $O \subset T \times P$,
- an initial marking $M_0 = (m_{01}.m_{02}, \dots, m_{0m})$.

There are several reasons why we chose Petri nets over other behavior models. First of all, many toolkits are available to easily construct these networks. Moreover, Petri nets can more easily be extended to incorporate useful functionality such as dealing with time. Lastly, there are many tools out there to create new petri nets that can be incorporated into a system in a modular fashion.

Petri nets have been extended in many ways in order to accommodate many different functionalities. The extension that attracts our attention the most is *generalized stochastic Petri nets* [8]. In this extension, there are two types of transitions: *immediate* and *timed* transitions. The generalized stochastic Petri net (GSPN) model can be described as a six-tuple:

$$GSPN = (P, T, I, O, M_0, \Lambda) \tag{2}$$

where (P, T, I, O, M_0) is the marked untimed PN underlying the SPN, and $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_n)$ is an array of (possibly marking dependent) firing rates associated with transitions.

Immediate transitions always have priority over timed transitions, and the likelihood of firing a timed transition is dependent on a parameter called the *firing rate* of the transition. This rate indicates the firing delay of the timed transition. This firing rate may be marking-dependent, so it should be written as $\lambda_i(M_j)$. The average firing delay of a transition t_i in marking M_j is $[\lambda_i(M_j)]_{-1}$.

A. Deadlines

The essential extension to the situations framework that is proposed in this paper adds an element of time to the system. This is needed to enable the system to deal with daily motion patterns. An important element without which the system cannot succeed is knowledge about how long actions are going to take. Only when this information is known to the agent (or system) it can be decided whether taking a certain action will result exceeding the deadline for the goal.

Central in our design is path planning to a *base place*, where a check is made if it is still possible to reach the (time constrained) goal. Then, we will compute for every transition that will not take the pedestrian to its goal, how much time it takes to get back to the base state. Then we can check whether the goal place is still accessible from the base state when a certain transition has been taken. We use this information to modify the timed transition rate, so pedestrians are more likely to choose the actions that leave them more time to reach their goal. This modification implies that the Petri net is no longer non-stationary.

B. Assumptions

The method we propose is based on various assumptions which have to be clarified. An important assumption is that the environment of the pedestrians is designed in such a way that it helps creating realistic behavior. This means that situations have to be defined in such a way that a pedestrian will walk into them and act in the appropriate way. In many cases, intuitively designed environments will cause the right behavior. For instance, in most train stations, the eating stands are placed in such a way that pedestrians will have to walk close to them in order to get to their destination.

Another assumption we make is that the pedestrians have to be at a certain place at a certain time. This makes the system more suitable for daily routine type situations rather than cases in which pedestrians are walking around without a proper goal. However, most situations can be described as having a deadline (e.g. eventually, most people have to go to bed), so this assumption is not necessarily very restrictive.

C. Path Planning

To initiate the stochastic Petri net, it has to be known how long an action will take. However, in many cases the duration of action is depended on the travelled distance. It is possible to do perform path-planning between often used places and store them before running the simulation. The travel distance can easily be computed using the *Dijkstra shortest path algorithm* [13]. In our system we can use this to compute the time from any place to the goal place (the source). This can be very useful when we would like to compute an estimate of how long a pedestrian will be busy with executing a behavior in a certain situation. But since the Petri nets are probabilistic, it will never be possible to give an exact prediction of the time it takes to execute a certain behavior.

In our implementation, we used the MASON multiagent simulation toolkit to create a 2 dimensional environment for our pedestrian agents.

IV. Results

A. Quantitative Experiment

It is very difficult to quantitatively establish whether lifelike behavior has been modeled. However, it is possible to check whether the mechanics of time planning work as predicted. In order to do this, we log the pedestrian's relative time when they arrive at their goal to check how much time they had left until their deadline. If the model works correctly, this time should roughly correlate to how the time probability function has been chosen. We will discuss the various functions we have used to model the probabilities over time.

B. Qualitative Experiment

Apart from quantitative analysis, we will also qualitatively judge the pedestrians' behavior. We will do this by comparing our modeled behavior with real-life behavior from recordings of Rotterdam airport. We have picked a couple of specific behaviors that we have modeled with our system.

1) The Behaviors:

Going to the toilet: In the videos, we observed that a typical behavior that manifests itself multiple times in the video material is that one person goes to the toilet, and another one waits until this person has come back. The Petri net used for this can be found in figure 2.

Standing in the queue for the check-in desk: One constant factor in the video material seemed to be the people lining up in front of the check-in desk. There are many ways to model this behavior with our framework. The situation area is chosen such that the pedestrians in it will together form an orderly line.

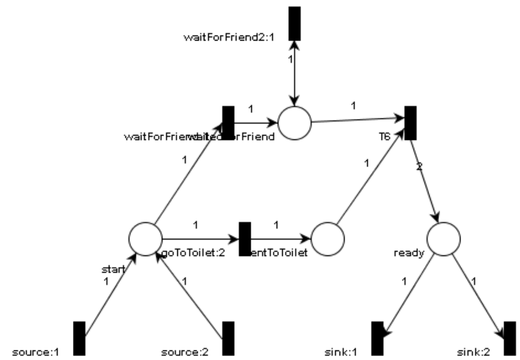

Fig. 2. Behavior model for the toilet situation. Transitions are given with black rectangles, places with white circles.

Checking in: When the pedestrians reach the end of the queue situation, they will enter the check-in situation, in which they will stand in front of the check-in desk for a little while, after which they are free to go again.

Lean Against Pillar: Another recurring behavior we saw is that people lean against the pillars in the hall. This is a type of idle behavior, a variation on the standing still behavior.

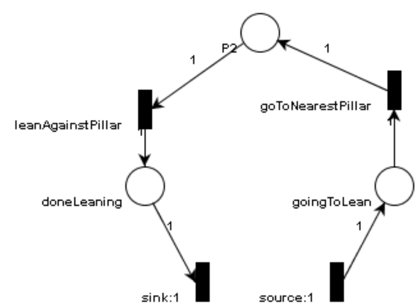

Fig. 3. Behavior model for the pillar situation

The Petri nets of those four situations are stationary; the firing rate is not a function of time. The duration of the behaviors for each situation is variable, which is stochastically decided. The four situations could be combined with a base state (see Fig. 4), where the decision is made which behavior model is activated. The probabilities of those transitions are a function of time. The probability for GoToGoal increases when the time approaches the departure time, while the probability of a transition to the behavior models ToiletSituation, StandInQueue and LeaningAgainstPillar decreases.

Figure 5 shows how those behaviors are combined in an environment which resembles the transit hall of Rotterdam airport. At the lower left a revolving door is visible, which is the place where agents enter the simulation. The upper right a number of check-in counters, at the upper left the security checkpoints towards the departure hall. The toilets are at the left, at the bottom a number of shops are present. This layout

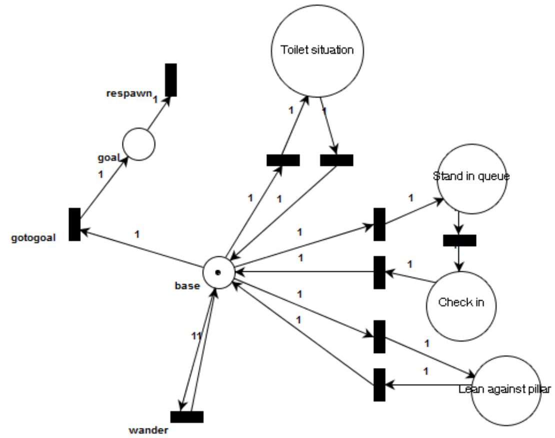

Fig. 4. Complete behavior model of a human in a transit hall

resembles the map as given in Fig. 1.

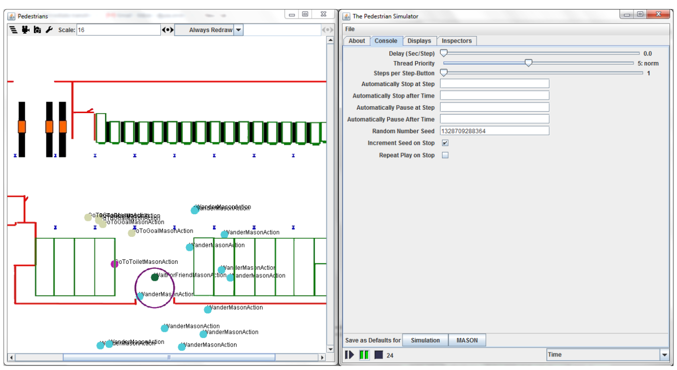

Fig. 5. Screenshot of the simulation environment with four different behaviors (Wander, GoToGoal, GoToToilet, WaitForFriend)

Figure 6 shows how the remaining time towards the security check is distributed when a large fraction of the pedestrians have decided to go the toilet (45% of the simulated agents). Detailed analysis of the logs showed that some pedestrians waited on each other and went to the security check in pairs. Furthermore, this shows that by specifying the probabilities for going to the pedestrians' goal, the probabilities for other actions are also affected.

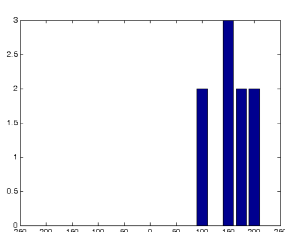

Fig. 6. Distribution of the remaining time towards the security check for pedestrians which went to the toilet.

V. Conclusion

In this paper an innovative method to model pedestrian behavior is presented. A time-augmented Petri net is used to make the behavior stochastic. The probability of starting a behavior is made a function of both the location and the time left. This framework is demonstrated on a departure hall scenario, where the time left is the departure time (in this case of an airplane). The conclusion can be made that this framework allows to generate a rich set of emergent motions, which makes it applicable to generate artificial testing data for suspicious behavior detection algorithms.

Acknowledgment

The authors would like to thank Philip Kerbusch for his input.

References

[1] B. E. Moore, S. Ali, R. Mehran, and M. Shah, "Visual crowd surveillance through a hydrodynamics lens," *Commun. ACM*, vol. 54, no. 12, pp. 64–73, Dec. 2011.

[2] F. E. Ritter, N. R. Shadbolt, D. Elliman, R. M. Young, F. Gobet, and G. D. Baxter, "Techniques for modeling human performance in synthetic environments: A supplementary review," Human Systems Information Analysis Center, Wright-Patterson Air Force Base, Ohio, State of the Art Report HSIAC SOAR 02-02, June 2002.

[3] J. C. Giordano, P. F. Reynolds, Jr., and D. C. Brogan, "Exploring the constraints of human behavior representation," in *Proceedings of the 36th conference on Winter simulation*, ser. WSC '04. Winter Simulation Conference, 2004, pp. 912–920.

[4] W. Shao and D. Terzopoulos, "Autonomous pedestrians," in *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ser. SCA '05. New York, NY, USA: ACM, 2005, pp. 19–28.

[5] M. Kallmann and D. Thalmann, "Modeling objects for interaction tasks," in *Proc. Eurographics Workshop on Animation and Simulation*, 1998, pp. 73–86.

[6] M. Sung, M. Gleicher, and S. Chenney, "Scalable behaviors for crowd simulation," *Comput. Graph. Forum*, vol. 23, no. 3, pp. 519–528, 2004.

[7] C. A. Petri, "Kommunikation mit automaten," Ph.D. dissertation, Universität Bonn, 1962, translated by C F Greene Jr, Applied Data Research Inc, Technical Report RADC-TR-65-377 Vol. 1 Supl. 1, Jan. 1966.

[8] F. Bause and P. S. Kritzinger, *Stochastic Petri nets - an introduction to the theory (2.ed.)*. Vieweg, 2002.

[9] R. P. P. Bonet, C.M. Llado and W. Knottenbelt, "Pipe v2.5: A petri net tool for performance modelling," in *Proc. 23rd Latin American Conference on Informatics (CLEI 2007)*, October 2007.

[10] J. Billington, S. Christensen, K. van Hee, E. Kindler, O. Kummer, L. Petrucci, R. Post, C. Stehno, and M. Weber, "The petri net markup language: Concepts, technology, and tools," in *Applications and Theory of Petri Nets 2003*, ser. Lecture Notes in Computer Science, W. van der Aalst and E. Best, Eds. Springer Berlin / Heidelberg, 2003, vol. 2679, pp. 483–505.

[11] J. Peterson, *Petri Nets and the Modelling of Systems*, ser. Series in Computer Science. MIT Press, 1981.

[12] M. Ajmone-Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis, *Modelling with Generalized Stochastic Petri Nets*, ser. Series in Parallel Computing. John Wiley and Sons, 1995.

[13] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.