From Sequential Decisions to Concurrent Processes: Advancing Human Navigation Models for Simulation

Sabine Timpf
Geoinformatics Group, Institute of Geography, University of Augsburg, Germany

Franziska Klügl
School of Science and Technology, University of Örebro, Sweden

Abstract

Computational models of human navigation have largely followed sequential decision paradigms, treating planning, perception, and movement as discrete models. In contrast, evidence from spatial cognition and neuroscience indicates that navigation arises from multiple interacting processes operating concurrently. Building on a previously proposed architecture, this paper advances a modelling framework that explicitly represents such concurrency and process interaction, and examines its implications for agent-based simulations of spatial decision-making in human navigation.

2012 ACM Subject Classification Computing methodologies → Model development and analysis; Computing methodologies → Intelligent agents; Computing methodologies → Multi-agent systems

Keywords and phrases agent-based simulation, modelling human navigation, computational process modelling

Digital Object Identifier 10.4230/LIPIcs.COSIT.2026.36

Category Short Paper

Supplementary Material
Software (Source Code): `https://github.com/GeosavvyAgents/concurrentNavigationProcesses`
archived at `swh:1:dir:8a5ea68e524352a605d85e33b2f7132b9c54bfce`

Funding *Sabine Timpf*: I gratefully acknowledge support by University of Örebro for my research stay in March 2026.

1 Human Navigation encompasses concurrent processes

Navigation in spatial environments cannot be reduced to a single decision-making process. Instead, navigation behaviour emerges from the interaction of multiple cognitive and sensorimotor processes, including perception, route planning, orientation, path following, and movement [17, 5, 24]. These processes operate at different spatial and temporal scales and continuously influence one another during navigation [1]. In addition, they are based on different representations of the spatial context at multiple levels of abstraction [27, 26].

For agent-based simulation, this implies that realistic models must account for the simultaneous and ongoing interplay between higher-level wayfinding and lower-level locomotion, rather than treating these processes as independent or sequential stages of the overall navigation process. However, most existing agent-based approaches to simulating navigation rely on architectures derived from the sense–plan–act paradigm [23]. These architectures typically impose a sequential control structure or introduce hierarchical dependencies between behavioural levels. As a result, they struggle to represent navigation as a dynamic, situated activity in which processes such as movement, re-orientation, and route-(re)planning occur in parallel and interact continuously. While hybrid models combine planning and reactive movement, they often do so within tightly coupled or centrally controlled frameworks. What

17th International Conference on Spatial Information Theory (COSIT 2026).
Editors: Sabine Timpf, Gabriele Filomena, Armand Kapaj, Rui Zhu, Nicholas Giudice, and Ed Manley; Article No. 36; pp. 36:1–36:8
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

remains largely unsupported is an explicit representation of navigation as a set of concurrent processes across scales that can operate independently while still interacting through the spatial context.

In this paper, we report on our progress in representing and simulating human navigation as interacting concurrent processes (routing, path following, orienting, moving) using shared states (places, paths, route) as interleaving principle. By shifting the focus from isolated decisions to interacting processes, we argue for a new generation of agent-based models for spatial behaviour that better reflect the complexity of human navigation. We expect to contribute to Spatial Information theory by introducing a conceptual and methodological framework for modelling human navigation as a set of explicitly represented, interacting concurrent processes, and demonstrating its implications for agent-based simulations of spatial decision-making.

The paper is structured as follows: After a short review of computational models of human navigation in section 2, we discuss the approaches for modelling concurrent processes and their interactions (section 3), and present our approach for modelling and simulating human navigation as loosely coupled processes coordinated through informational states (section 4). We then discuss the current state of the research (section 5) and present our (preliminary) conclusions as well as some future work in section 6.

2 Modelling Human Navigation

Research on human navigation spans cognitive science, geography, robotics, and simulation, yet remains fragmented with respect to computational process modelling. Reviews of spatial memory (e.g., [16]) emphasize internal representations, while pedestrian and crowd models (e.g., [12, 18]) focus on movement dynamics and local interactions. Behavioural accounts [8] describe navigation as a multi-stage activity, but are rarely formalized computationally. In contrast, robotics approaches (e.g., [25]) provide explicit computational frameworks, yet typically rely on sequential sense–plan–act architectures. Agent-based models integrate cognitive and environmental factors (e.g., [3, 6]), but generally adopt modular or staged designs. Across these strands, navigation is decomposed into components, but not modelled as a system of interacting, concurrent processes.

While many computational models acknowledge that behaviour emerges from multiple interacting processes, they rarely treat these processes as explicitly concurrent at the architectural level. Early cognitive models such as NAVIGATOR [9] and PLAN [2] decompose navigation into functional components, but implement them largely as sequential reasoning steps without continuous interaction with locomotion. More recent pedestrian simulations – such as those by Kneidl et al. [14], Gath-Morad et al. [7], and Kielar et al. [13] – introduce multi-layered structures that separate high-level routing from low-level movement. These approaches implicitly recognize concurrency, for example by combining path planning with reactive collision avoidance, yet interactions remain tightly coupled or hierarchically controlled.

The work of Torrens [28] is particularly notable in this context. His agent-based framework integrates path planning, environmental perception, and locomotion within a unified system, enabling dynamic adaptation to context and other agents. However, even here, concurrency is realized through tightly integrated modules within a single decision framework rather than as independent, asynchronously operating processes.

In contrast, the approach proposed in this paper treats concurrency as a first-class modelling principle: navigation is represented as a set of loosely coupled processes operating in parallel and interacting through shared state, rather than through centralized control or sequential coordination. This distinction enables a more transparent and flexible representation of the interdependencies that characterize human navigation.

The spatial scale addressed in this paper is the local scale of individual navigation. The framework is concerned with agents moving through environments such as buildings, campuses, parks, street networks, or neighbourhood-scale settings, where navigation involves route following, orientation, perception of local cues, and possible adjustment of movement or route-related decisions. It is not primarily aimed at regional transport modelling, large-scale activity scheduling, or aggregate travel-demand forecasting, although such models could provide higher-level constraints or destinations.

3 Modelling concurrent processes and their interactions

Concurrent processes in core computer science are commonly modelled using formalisms from distributed systems [29]. There, a basic distinction is made between message-based and shared-memory-based paradigms. In message-based modelling languages, such as Communicating Sequential Processes or the π-calculus, processes exchange information through explicit communication, whereas shared-memory approaches coordinate processes through common variables. Communicating automata similarly model interaction by showing how messages or events trigger transitions and produce further events. For modelling concurrent processes within one agent's decision-making, however, the key issue is not the communication mechanism itself, but the dependencies between processes: information needs, enabling conditions, and required updates. In a message-based approach, process A sends information to process B; in a shared-memory approach, process A writes information to a variable that process B monitors. In both cases, the relevant contribution is to make inter-process information dependencies explicit.

Thus, for the level of abstraction that is relevant here, the shared memory-based approach appears to be more appropriate. Clearly, we ignore issues related to performance, robustness or the complexity of synchronizing shared access avoiding race conditions. At this level of abstraction, the major languages to capture process behaviour and interaction mostly associated with state changes with explicit conditions (events) are Petri-Nets and more basic statechart-like languages that allow to formulate more condition-driven coordination between concurrent processes.

Petri nets [22], in particular in their condition–event interpretation, model system behaviour in terms of conditions represented by places and events represented by transitions, with tokens placed in condition nodes indicating the current state of the system. A transition is enabled when all required input conditions are satisfied (have a token), making dependencies between process steps explicit. While this provides a precise representation of causal relationships, it is not always easy to distinguish clearly how different concurrent processes develop due to the tight coupling between overall system state representation and conditions for process progression.

Another language that occurs in the context of modelling concurrent processes – more for verification – is TLA+[15]. It is a state-based specification language based on temporal logic that characterizes overall system behaviour in terms of transitions between states given logical constraints. Transitions happen based on actions from different concurrent processes. While it can be used to model concurrent processes, it does not provide an explicit notion of process structure, instead it captures concurrency implicitly through nondeterministic interleaving of enabled actions. Yet, we need explicit representation of interdependent processes and their progression, not just constraints on global behaviour.

Statecharts extend classical automata-based models by introducing hierarchical structuring, explicit support for formulating multiple loosely coupled processes that advance concurrently combined with condition-based transitions between different (sub-)process states. Those transitions can be guarded by conditions using the overall agent's informational state, enabling the explicit specification of dependencies between processes. Variants of statechart-based modeling are widely used in software engineering practice including the original formulation by D. Harel [10] to UML state machines [19].

A related question refers to how the agent's informational state – representing the shared memory for the concurrent processes – can be represented. In a simple object-oriented agent implementation, this may be represented as agent-level variables that are read and updated by the different concurrent internal processes. More elaborate agent architectures may use a belief base, in which information about the agent, its environment, and relevant contextual knowledge can be represented explicitly, often using some logic-based language as in BDI-type of architectures [21]. A further alternative is a blackboard architecture, where the shared state is more like a common problem-solving workspace [11]. Tuple-space models, such as Linda, provide a related coordination mechanism in which processes interact through a shared repository of structured data items [20]. Finally, publish–subscribe infrastructures (as in ROS) can be used when processes should remain more decoupled: rather than directly accessing shared variables, processes publish updates and subscribe to information types or events relevant to their own progression. These alternatives differ mainly in how explicit, structured, and decoupled the shared informational state is. Agent-level variables are simple and efficient but provide little semantic structure. Belief bases and blackboard systems make the informational content more explicit and extensible. Tuple spaces and publish–subscribe mechanisms further decouple information producers and consumers, but may introduce additional architectural complexity.

4 Proposed Approach for Simulating Concurrent Processes for Human Navigation

Our approach is conceptually related to blackboard architectures [11], in which multiple independent processes interact through a shared state without a fixed control sequence. While such architectures are no longer widely used as a software engineering paradigm, their core principle of state-mediated, opportunistic interaction remains influential in contemporary models of cognition and distributed systems.

We conceptualize the core navigation of pedestrians as consisting of at least four processes, i.e., routing, path following, orienting (also called self-localization), and walking, as originally suggested by Downs and Stea [4]. Figure 1 shows the conceptual model underlying each of the core navigation processes. It demonstrates how routing, path following, orienting, and walking each carry out their own reasoning, from surface-level motion, called walking, to abstract graph routing. This also implies different spatial representations for each reasoning process, shown on the right side of figure 1.

The orienting process stores orientation information in a memory structure, which is also used to compare locations, and checks if the goal place is reached. In the walking process, the agent moves from a starting location to a goal location on a surface. The path following process is based on concepts of places, path segments as connections between places, and the

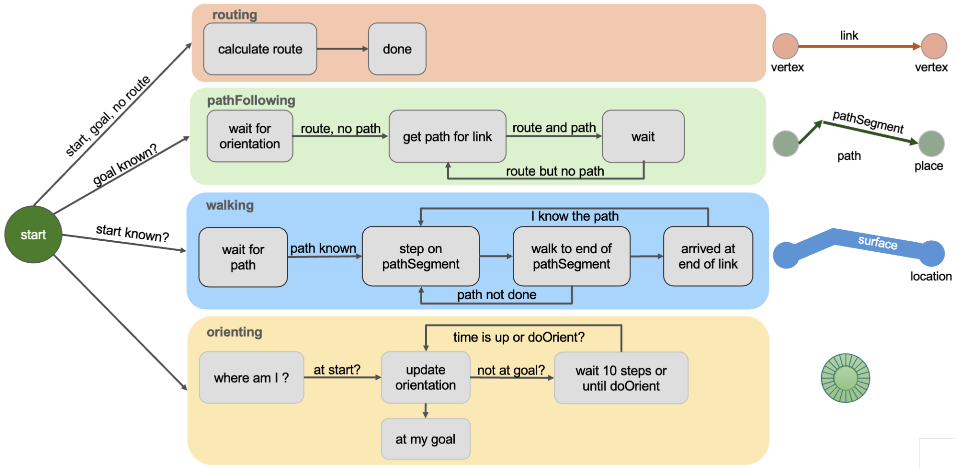

Figure 1 Conceptual model of behavioural processes and corresponding spatial representations.

notion of path as a connected series of path segments; finally, the routing process uses vertices, links, and the notion of a route. The path following process accesses metric information in its places and segments while the routing happens on a graph (seen as a purely topological structure). Locations are abstracted to places, some of which are abstracted to vertices. The surface in the walking process is simplified to a segment and a list of segments is abstracted to a link.

The conceptual model mostly follows the ideas of Downs and Stea [4] with the exception of the process of *route monitoring*. In this instance of our model, *route monitoring* consists of determining if the current position is consistent with being on the planned path. We suggest to integrate this process into the orienting process, with the monitoring being the comparison between the current and the planned orientation.

A key conceptual step is the stratification of behavioural processes. This separation is maintained by restricting interactions to shared agent or process states. Navigation is thus decomposed into four concurrent, theory-grounded sub-processes, forming the basis of our simulation.

Conceptualizing navigation as a set of concurrent, interdependent processes requires appropriate computational support. We implement the model in SeSAm[1], which natively supports concurrency through multiple statecharts, termed *reasoning engines*. Each reasoning engine encodes a single behavioural process at the agent level. Multiple engines can be active concurrently, enabling loosely coupled processes without enforced synchronization. The architecture is explicitly designed for agents engaging in concurrent activities (e.g., negotiating while executing a task).

The execution of concurrent processes is technically interleaved across atomic update cycles in randomized order, leaving process scheduling uncontrolled. This necessitates explicit coordination via shared informational states, aligning well with our design principle of process separation. The model code is available online on GitHub[2].

Figure 2 shows the four reasoning engines and the six shared state variables mediating their interactions. Processes influence one another indirectly via these variables (e.g., path-related information), avoiding centralized control. This preserves behavioural modularity.

[1] `http://simsesam.org/`
[2] `https://github.com/GeosavvyAgents/concurrentNavigationProcesses`

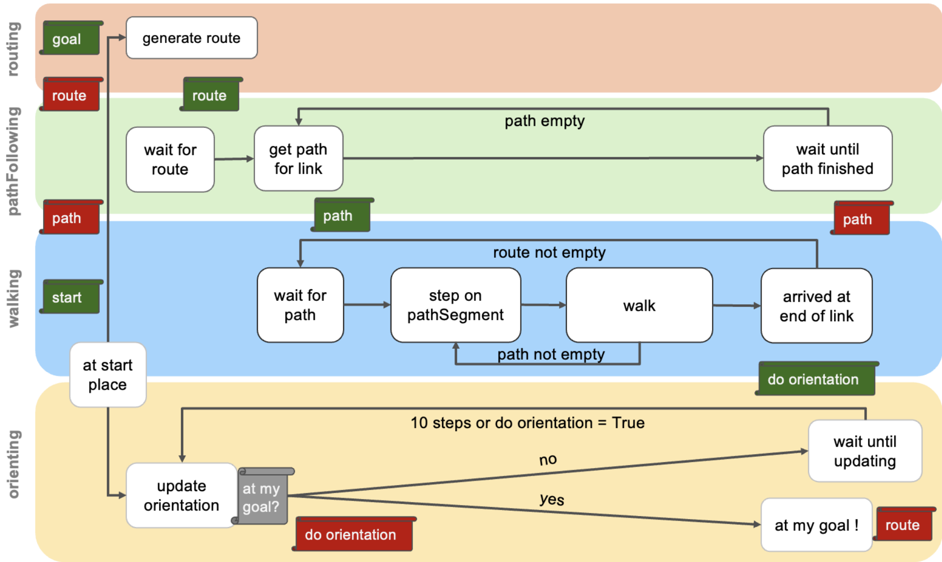

Figure 2 Four processes and their interaction with six state variables shown in green (state is true) and in orange (state is false).

Processes are conditionally activated and, once active, proceed concurrently. Each reasoning process can be replaced by alternative models or algorithms, provided they adhere to the same interface via shared state variables.

5 Discussion

The proposed framework reconceptualizes navigation as a set of concurrent processes coordinated through shared informational states, rather than sequential decision stages. This enables a more faithful representation of navigation as a continuous, situated activity in which perception, orientation, and movement interact dynamically. By decoupling processes and making dependencies explicit via shared state variables, the approach increases transparency compared to tightly integrated architectures.

The use of loosely coupled reasoning engines introduces a form of modularity that supports substitutability. Individual processes – such as routing or orientation – can be replaced by alternative models without requiring changes to the overall architecture, provided they conform to the shared state interface. This suggests a pathway toward reusable modelling patterns for navigation behaviour.

At the same time, this design introduces new challenges. The coordination of processes is no longer enforced by a central control structure but emerges from the correct design of shared state variables and enabling conditions. As a result, model validity depends critically on identifying appropriate informational states and their semantics, which remains a domain-specific and non-trivial task.

Finally, while the framework captures concurrency at the architectural level, the current implementation remains a proof of concept. The behavioural richness of individual processes is still limited, and their empirical grounding requires further investigation. Thus, the main contribution lies in the structuring of interaction rather than in the sophistication of the underlying behavioural models.

6 Conclusions and future work

This work-in-progress paper introduced a conceptual and computational framework for modelling human navigation as a set of interacting concurrent processes. By treating concurrency as a first-class modelling principle and coordinating processes through shared informational states, the approach provides an alternative to sequential and hierarchically structured agent architectures. The implementation in SeSAm demonstrates the feasibility of this design and highlights its potential for representing navigation as a dynamic and distributed activity.

Future work will proceed along three main directions. First, the individual reasoning processes will be extended with empirically grounded models from spatial cognition, enabling more realistic behaviour. Second, we aim to systematically identify and formalize the shared informational states required for coordinating concurrent processes, with the goal of developing reusable modelling patterns. Third, the framework will be evaluated through simulation experiments, including comparisons with sequential architectures to assess its impact on behavioural realism and explanatory power.

In the longer term, the proposed approach may contribute to a broader rethinking of agent-based models of spatial behaviour, in which decision-making is no longer conceptualized as a sequence of choices but as the emergent outcome of interacting processes operating across multiple temporal and representational scales.

References

1 Negar Alinaghi, Ioannis Giannopoulos, Markus Kattenbeck, and Martin Raubal. Decoding wayfinding: analyzing wayfinding processes in the outdoor environment. *Int. Journal of Geographical Information Science*, pages 1–31, March 2025. doi:10.1080/13658816.2025.2473599.

2 Eric Chown, Stephen Kaplan, and David Kortenkamp. Prototypes, location and associative networks (plan): Towards a unified theory of cognitive mapping. *Cognitive Science*, 19(1):1–52, 1995. doi:10.1016/0364-0213(95)90017-9.

3 Andrew T. Crooks, Christopher J. E. Castle, and Michael Batty. Key challenges in agent-based modelling for geo-spatial simulation. *Computers, Environment and Urban Systems*, 32(6):417–430, 2008. doi:10.1016/j.compenvurbsys.2008.09.004.

4 R.M. Downs and D. Stea. *Maps in Minds.* Harper & Row, New York, US, 1977.

5 Anna Charisse Farr, Tristan Kleinschmidt, Prasad Yarlagadda, and Kerrie Mengersen. Wayfinding: A simple concept, a complex process. *Transport Reviews*, 32(6):715–743, November 2012. doi:10.1080/01441647.2012.712555.

6 Gabriele Filomena and Judith A. Verstegen. Modelling the effect of landmarks on pedestrian dynamics in urban environments. *Computers, Environment and Urban Systems*, 86:101573, 2021. doi:10.1016/j.compenvurbsys.2020.101573.

7 M. Gath-Morad, E. Ronchi, and R. Lovreglio. A multi-agent approach for modelling pedestrian dynamics: A case study. In *Proceedings of the International Conference on Pedestrian and Evacuation Dynamics*, 2013.

8 Reginald G. Golledge. *Wayfinding Behavior: Cognitive Mapping and Other Spatial Processes.* Johns Hopkins University Press, Baltimore, 1999.

9 S. Gopal, Roberta L. Klatzky, and Thomas R. Smith. Navigator: A psychologically based model of environmental learning through navigation. In *Proceedings of the Spatial Information Theory Conference*, 1989.

10 D. Harel. Statecharts: a visual formalism for complex systems. *Science of Computer Programming*, 8(3):231–274, 1987. doi:10.1016/0167-6423(87)90035-9.

11 Barbara Hayes-Roth. A blackboard architecture for control. *Artificial Intelligence*, 26(3):251–321, July 1985. doi:10.1016/0004-3702(85)90063-3.

12 Dirk Helbing and Peter Molnár. Social force model for pedestrian dynamics. *Physical Review E*, 51(5):4282–4286, 1995. doi:10.1103/PhysRevE.51.4282.

13 Peter M. Kielar and André Borrmann. Spice: a cognitive agent framework for computational crowd simulations in complex environments. *Autonomous Agents and Multi-Agent Systems*, 32(3):387–416, 2018. doi:10.1007/s10458-018-9383-2.

14 Angelika Kneidl, André Borrmann, and Dirk Hartmann. Generation and use of sparse navigation graphs for microscopic pedestrian simulation models. In *Advances in Intelligent Systems and Computing*, pages 189–198, 2012. doi:10.1007/978-3-642-29709-0_18.

15 L. Lamport. *Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers*. Addison Wesley, Boston, 2002.

16 Thomas Madl, Ke Chen, Daniela Montaldi, and Robert Trappl. Computational cognitive models of spatial memory in navigation space: A review. *Neural Networks*, 65:18–43, 2015. doi:10.1016/j.neunet.2015.01.008.

17 D.R. Montello. Navigation. In P. Shah and A. Miyake, editors, *Handbook of visuospatial cognition*, pages 257–294. Cambridge University Press, Cambridge, 2005.

18 Mehdi Moussaïd, Mubbasir Kapadia, Trevor Thrash, Robert W. Sumner, Markus Gross, Dirk Helbing, and Christoph Hölscher. Crowd behaviour during high-stress evacuations in an immersive virtual environment. *Journal of The Royal Society Interface*, 13(122):20160414, 2016. doi:10.1098/rsif.2016.0414.

19 Christoph Oechslein, Franziska Klügl, Rainer Herrler, and Frank Puppe. Uml for behavior-oriented multi-agent simulations. In Barbara Dunin-Keplicz and Edward Nawarecki, editors, *From Theory to Practice in Multi-Agent Systems*, pages 217–226, Berlin, Heidelberg, 2002. Springer. doi:10.1007/3-540-45941-3_23.

20 A. Omicini, F. Zambonelli, and M. Klusch, editors. *Coordination of Internet Agents: Models, Technologies, and Applications*. Springer, 2014.

21 A. S. Rao and M. P. Georgeff. BDI agents: From theory to practice. In *Proc. of the 1st Int. Conf. on Multiagent Systems, ICMAS95*, pages 312–319, 1995.

22 W. Reisig. *Understanding Petri Nets: Modeling Techniques, Analysis Methods, Case Studies*. Springer, Berlin, Heidelberg, 2013.

23 S. Russell and P. Norvig. *Artificial Intelligence – A Modern Approach*. Pearson, Harlow, 4 edition, 2022.

24 Angela Schwering, Jakub Krukar, Rui Li, Vanessa Joy Anacta, and Stefan Fuest. Wayfinding Through Orientation. *Spatial Cognition & Computation*, 17(4):273–303, October 2017. doi:10.1080/13875868.2017.1322597.

25 Sebastian Thrun. Robotic mapping: A survey. In *Exploring Artificial Intelligence in the New Millennium*, pages 1–35. Morgan Kaufmann, 2002.

26 Sabine Timpf and Werner Kuhn. Granularity Transformations in Wayfinding. In Christian Freksa, Wilfried Brauer, Christopher Habel, and Karl Friedrich Wender, editors, *Spatial Cognition III, Routes and Navigation, Human Memory and Learning, Spatial Representation and Spatial Learning*, Lecture Notes in Computer Science, pages 77–88. Springer, 2003. doi:10.1007/3-540-45004-1_5.

27 Sabine Timpf, Gary S. Volta, David W. Pollock, and Max J. Egenhofer. A conceptual model of wayfinding using multiple levels of abstractions. In A.U. Frank, I. Campari, and U Formentini, editors, *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, volume 639 of *LNCS*, pages 348–367. Springer, Heidelberg Berlin, 1992. doi:10.1007/3-540-55966-3_21.

28 Paul M. Torrens. Moving agent pedestrians through space and time. *Annals of the Association of American Geographers*, 102(1):35–66, 2012. doi:10.1080/00045608.2011.595658.

29 M. van Steen and A. Tanenbaum. *Distributed Systems*. distributed.systems.net, 4th edition, 2023.