Spatial Programming for Environmental Monitoring

Josh Millar*
Imperial College London

Ryan Gibb*
University of Cambridge

Roy Ang
University of Cambridge

Hamed Haddadi
Imperial College London

Anil Madhavapeddy
University of Cambridge

Abstract

Large-scale environmental monitoring demands real-time, spatially-aware coordination across distributed networks. However, existing distributed computing models poorly capture spatial structure, hindering dynamic collaboration and fine-grained access control. We argue that space must be treated as a first-class concept in programming models for these systems based on bigraphs – a formalism that explicitly models spatial arrangements, data movement, and access policies, while supporting real-time reconfiguration and localised reasoning. This approach facilitates secure, composable, and dynamically verifiable coordination across geographically distributed nodes and organisations, paving the way for scalable, responsive environmental networks.

1 The Case for Spatial Programming

The world's most pressing challenges – from climate change to biodiversity loss – demand large-scale, spatially-distributed monitoring and reasoning. Global environmental monitoring networks already deploy millions of nodes across vast, heterogeneous regions: carbon flux towers across the Amazon basin [5], acoustic monitoring across the Great Barrier Reef [3], marine salinity monitors on drifting buoys [8], and even satellites orbiting the Earth [9]. These networks are increasingly expected to support real-time decision-making and cross-organisational collaboration at unprecedented scale [2]. Yet, our current dominant computing paradigms are poorly suited to managing inherently *spatial* arrangements.

Traditional models of distributed computing prioritise message passing and connectivity, but these abstractions fail to capture the spatial dynamics of environmental networks. Typically, data is captured locally, shipped to central servers, and made available after processing – often following complex transformations and lengthy access agreements. This model introduces delays and makes real-time coordination practically unachievable. For example, if multiple conservation organisations wish to jointly manage a transboundary habitat, they are typically forced to manually share and replicate data repositories, design custom APIs, and construct static access control schemes. These mechanisms are also poorly equipped to handle fine-grained, location-aware reasoning – such as selectively sharing live data only within active patrol zones, or automatically updating permissions as data traverses spatial boundaries.

What is needed is a programming model that treats space as a first-class dimension, enabling developers to explicitly describe spatial layouts, boundaries, and location-dependent policies. More importantly, it would also enable dynamic spatial reasoning: determining who can see which data, at what times, and in which places should be an adaptable, context-driven property of the platform itself. Without this capability, programming large-scale monitoring infrastructures remains fundamentally mismatched to the spatial complexity of the real-world phenomena they aim to monitor.

We argue that spatial programming – explicitly embedding spatial structure and movement into software design – is a necessary step towards building more robust, scalable, and responsive environmental networks. A spatial programming approach allows us to reason about and control how nodes, agents, and data interact across space, not just across the network. Importantly, this opens the door to spatial access control: the ability to govern data flow based on spatial structure in real-time. Instead of constructing bespoke data pipelines and coarse-grained sharing agreements, we can use a universal, composable model of space to scope collaboration, data visibility, and policy enforcement directly to the spatial regions where they are relevant.

Bigraphs provide a model of the space and motion of communication agents [4], capturing spatial, organisational, and communication relationships within a single formalism. Formally, a bigraph consists of two orthogonal structures over the same set of typed entities (Fig. 1). The *place* graph is a rooted forest encoding nested spatial hierarchy, with roots naturally modelling distinct adjacent *regions*. The *link* graph is a hypergraph connecting entity *ports*, encoding non-spatial relationships. For instance, a network or a data flow link can be modelled as a hyperedge joining multiple ports (*e.g.*, data flow paths between organisations, or even monitoring routes such as ecological corridors). Joining the link graph with the place graph enables the modelling of scenarios where actions or policies depend on both location and non-spatial connections. Bigraphs include *closed links* – complete hyperedges

*Both authors contributed equally to this research.

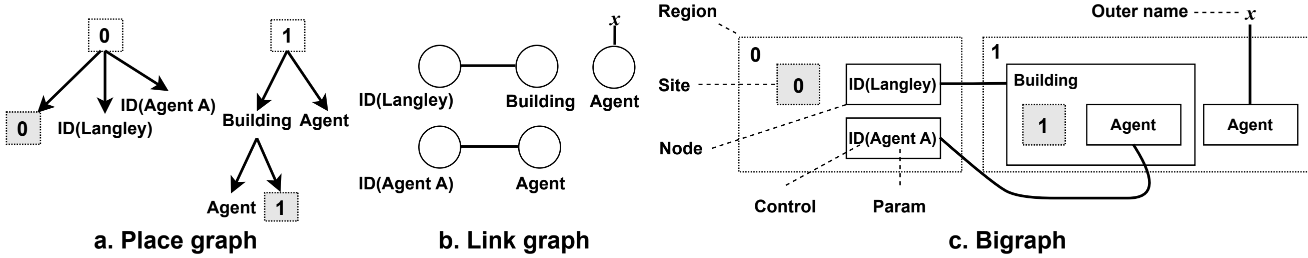

Figure 1: An illustrative *bigraph*, constructed from its underlying *place* and *link* graphs.

connecting nodes – alongside *open links*, incomplete hyperedges ending in *outer names* (*e.g.*, x in Fig. 1). Each node is associated with a *control*, defining its type and interface. Fig. 2 outlines how a bigraph models an urban monitoring network, capturing various data sources and 'applications' or data processing pipelines.

Reaction rules allow transformations over the bigraph, encoding policy (who can connect to what node), and providing a mechanism to program dynamic changes to physical environments in real-time. For example, a rule can model data moving between connected organisations, automatically severing links when crossing a spatial boundary.

```
react shutdown_nodes =
/x (Room.(Users.() || Node{x} || rest))
--> Room.(rest);

begin brs
init ...;
rules = [{shutdown_nodes}, {...}];
end
```

Listing 1: A reaction rule: all nodes in `Room` are shut down when no users are present.

While tooling exists to compute the transition systems of Bigraphical Reactive Systems, such as BigraphER [6], current approaches require programming directly in low-level formalisms that are unwieldy for expressing high-level spatial concepts. Although languages like Verse [1] are emerging for spatial programming in 3D game engines, they lack the foundational spatial framework based on hierarchy of spatial containment that bigraphs provide.

We envision a higher-level spatial programming language that treats spatial modelling as a first-class concern, compiling down to bigraphical representations for execution and verification. Such a language would enable declarative spatial policies, automatic data flow management across organisational boundaries, and real-time adaptation to changing conditions. Rather than retrofitting spatial reasoning onto existing systems, we can embed it directly into the programming model – making space-aware coordination natural and composable.

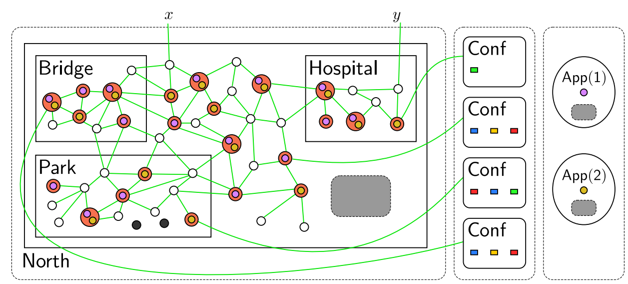

Figure 2: Bigraphical model of an urban network – the three regions in the graph represent the physical layer (nodes in space), the data layer (their gathered data), and the application layer (a data pipeline built atop the network) [7].

References

[1] Lennart Augustsson, Joachim Breitner, Koen Claessen, Ranjit Jhala, Simon Peyton Jones, Olin Shivers, Guy L. Steele Jr., and Tim Sweeney. 2023. The Verse Calculus: A Core Calculus for Deterministic Functional Logic Programming. *Proc. ACM Program. Lang.* 7, ICFP, Article 203 (Aug. 2023), 31 pages. https://doi.org/10.1145/3607845

[2] Stephanie E. Hampton, Carly A. Strasser, Joshua J. Tewksbury, Wendy K. Gram, Amber E. Budden, Archer L. Batcheller, Clifford S. Duke, and John H. Porter. 2013. Big data and the future of ecology. *Frontiers in Ecology and the Environment* 11, 3 (2013), 156–162. https://doi.org/10.1890/120103

[3] Ashlee Lillis, Amy Apprill, Justin J. Suca, Cynthia Becker, Joel K. Llopiz, and T. Aran Mooney. 2018. Soundscapes influence the settlement of the common Caribbean coral *Porites astreoides* irrespective of light conditions. *Royal Society Open Science* 5, 12 (2018), 181358. https://doi.org/10.1098/rsos.181358

[4] Robin Milner. 2008. The Space and Motion of Communicating Agents. (December 2008). https://www.cl.cam.ac.uk/archive/rm135/Bigraphs-draft.pdf Unpublished manuscript, University of Cambridge.

[5] Natalia Restrepo-Coupe, Humberto R. da Rocha, Lucy R. Hutyra, et al. 2013. What drives the seasonality of photosynthesis across the Amazon basin? A cross-site analysis of eddy flux tower measurements from the Brasil flux network. *Agricultural and Forest Meteorology* 182-183 (2013), 128–144.

[6] Michele Sevegnani and Muffy Calder. 2016. BigraphER: Rewriting and Analysis Engine for Bigraphs. In *Computer Aided Verification - 28th International Conference, CAV 2016, Toronto, ON, Canada, July 17-23, 2016, Proceedings, Part II.* 494–501. https://doi.org/10.1007/978-3-319-

41540-6_27
[7] Michele Sevegnani, Milan Kabac, Muffy Calder, and Julie McCann. 2018. Modelling and Verification of Large-Scale Sensor Network Infrastructures. In *2018 23rd International Conference on Engineering of Complex Computer Systems (ICECCS)*. 71–81. https://doi.org/10.1109/ICECCS2018.2018.00016
[8] Annie P. S. Wong, Susan E. Wijffels, Stephen C. Riser, Sylvie Pouliquen, Shigeki Hosoda, Dean Roemmich, John Gilson, Gregory C. Johnson, Kim Martini, David J. Murphy, et al. 2020. Argo Data 1999–2019: Two Million Temperature–Salinity Profiles and Subsurface Velocity Observations From a Global Array of Profiling Floats. *Frontiers in Marine Science* 7 (2020), 700. https://doi.org/10.3389/fmars.2020.00700
[9] Michael A. Wulder, J. G. Masek, Warren B. Cohen, Thomas R. Loveland, and Curtis E. Woodcock. 2012. Opening the archive: how free data has enabled the science and monitoring promise of Landsat. *Remote Sensing of Environment* 122 (2012), 2–10. https://doi.org/10.1016/j.rse.2012.01.010