



WIKIPEDIA
The Free Encyclopedia

Ambient calculus

In computer science, the **ambient calculus** is a process calculus devised by Luca Cardelli and Andrew D. Gordon in 1998, and used to describe and theorise about concurrent systems that include *mobility*. Here *mobility* means both computation carried out on mobile devices (*i.e.* networks that have a dynamic topology), and mobile computation (*i.e.* executable code that is able to move around the network). The ambient calculus provides a unified framework for modeling both kinds of mobility.^[1] It is used to model interactions in such concurrent systems as the Internet.

Since its inception, the ambient calculus has grown into a family of closely related ambient calculi.

Informal description

Ambients

The fundamental primitive of the ambient calculus is the **ambient**. An ambient is informally defined as a *bounded* place in which computation can occur. The notion of boundaries is considered key to representing mobility, since a boundary defines a contained computational agent that can be moved in its entirety.^[1] Examples of ambients include:

- a web page (bounded by a file)
- a virtual address space (bounded by an addressing range)
- a Unix file system (bounded within a physical volume)
- a single data object (bounded by “self”)
- a laptop (bounded by its case and data ports)

The key properties of ambients within the Ambient calculus are:

- Ambients have names, which are used to control access to the ambient.
- Ambients can be nested inside other ambients (representing, for example, administrative domains)
- Ambients can be moved as a whole.

Operations

Computation is represented as the crossing of boundaries, *i.e.* the movement of ambients. There are four basic operations (or capabilities) on ambients:^[1]

- *in m. P* instructs the surrounding ambient to enter some sibling ambient *m*, and then proceed as *P*
- *out m. P* instructs the surrounding ambient to exit its parent ambient *m*
- *open m. P* instructs the surrounding ambient to dissolve the boundary of an ambient *m* located at the same level

- *copy* *m*. makes any number of copies of something *m*

The ambient calculus provides a reduction semantics that formally defines what the results of these operations are.

Communication *within* (i.e. local to) an ambient is anonymous and asynchronous. Output actions release names or capabilities into the surrounding ambient. Input actions capture a value from the ambient, and bind it to a variable. *Non-local* I/O can be represented in terms of these local communications actions by a variety of means. One approach is to use mobile “messenger” agents that carry a message from one ambient to another (using the capabilities described above). Another approach is to emulate channel-based communications by modeling a channel in terms of ambients and operations on those ambients.^[1] The three basic ambient primitives, namely **in**, **out**, and **open** are expressive enough to simulate name-passing channels in the π -calculus.

See also

- [Lambda calculus](#)
- [Mobile membranes](#)
- [Type theory](#)
- [API-Calculus](#)

References

1. Cardelli, L.; A.D. Gordon. "Mobile Ambients". *Proceedings of the First International Conference on Foundations of Software Science and Computation Structure (March 28 - April 4, 1998)*. M. Nivat, ed. *Lecture Notes in Computer Science*. Springer-Verlag. **1378**: 140–155.

External links

- [Mobile Computational Ambients \(http://lucacardelli.name/Papers/MobileAmbients.A4.pdf\)](http://lucacardelli.name/Papers/MobileAmbients.A4.pdf) by [Luca Cardelli](#)
-

Retrieved from "https://en.wikipedia.org/w/index.php?title=Ambient_calculus&oldid=1142447652"

▪