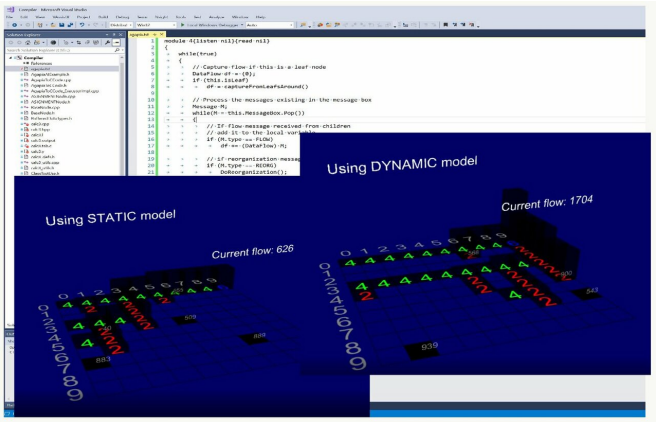
A new model introduces the concept of adaptive virtual organisms (VOs)

31 January 2019, by Ingrid Fadelli



Credit: Paduraru & Stefanescu.

Researchers at the University of Bucharest have recently developed a compositional model for complex hardware-software binding. Their model, outlined in a paper pre-published on arXiv, introduces the notion of a "virtual organism" (VO) that resides somewhere between slightly reconfigurable hardware agents and abstract, intelligent and adaptive software agents.

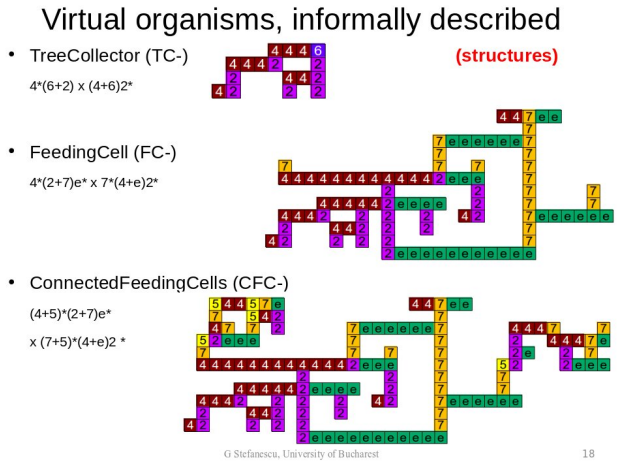
The relationship between a structure and the function it runs is a topic of interest in several fields, including computer science (hardware vs software), biology (organs vs function) and psychology (body vs mind). Ciprian Paduraru and Gheorghe Stefanescu, the two researchers who carried out the recent study, set out to investigate the relationship between hardware and software in computer science, particularly in the context of robotics, AI-hardware, IoT and other recent technological advances.

"Actions of a sequential computation can be easily controlled, but are often difficult to parallelize, while a natively distributed, parallel application is usually difficult to control," Stefanescu told TechXplore.

"To find a robust, mixed setting, we previously introduced a space-time duality-based model (rv-IS) and a DSL structured programming language (Agapia)."

Agapia is a domain-specific language (DSL) used for programming interactive systems, where dataflow and control flow structures can be freely mixed. Its compiler can currently produce high performance computing (HPC) runs, within either MPI or OpenMP environments.

Agapia's operational semantics are described by 2-D structures, with one dimension for time and one dimension for space. To effectively deal with space-time constraints, Paduraru and Stefanescu devised a new way to define regular 2-D patterns over arbitrary shape words. This allowed them to extend their model, giving it more dimensions for space.

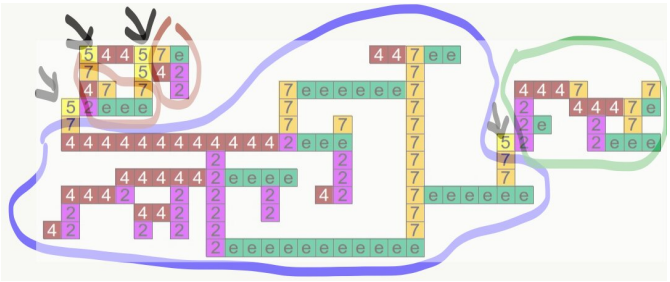


Credit: Paduraru & Stefanescu.

"When I presented the model to Gul Agha at the University of Illinois Urbana Champaign in the summer of 2015, and asked him whether he thought it was a good model for agents, he pointed

out a missing feature: adaptation," Stefanescu said. "Later on, we realized that structural adaptation may be easily included, allowing the system to change, at runtime, its structure to another structure from a class of allowed patterns."

Stefanescu and Paduraru hope that once completed, their model will enable a new type of "assembly language" that bridges distributed software and hardware applications. One of the key contributions of their study is that it introduces the concept of "virtual organisms," which have a structure that reflects hardware capabilities and run low-level functions, implementing the software requirements.



Credit: Paduraru & Stefanescu.

"A class of 2-D virtual organisms (VOs) is defined by a combination of structural and functional specifications," Stefanescu said. "The structural information is provided by a regular 2-D pattern, describing the allowed structures for the placement of computing nodes. The neighbouring nodes communicate via their shared interfaces. The nodes with interfaces on the external VO's border assure the interaction with the environment. Moreover, the nodes on the external border have a key role in controlling the spatial composition of VOs, allowing the virtual organisms to aggregate into larger organisms."

As a VO evolves, it might change its structure, adding or deleting nodes via reconfiguration, provided that the new structure is in the same class as its current one. Explicit creation and deletion operators can also be specified. The basic functionalities supported by a specific class of VOs are implemented by the VO's network of computing nodes. These include functions supported by the nodes and the communication enabled by the VO's structure.

"A managing program controls which functions are executed, where and how they interfere," Stefanescu explained. "Among the supported basic functionalities, there are special ones dedicated to adaptation: They decide if reconfiguration, addition or deletion of nodes are necessary, as well as when and how they are performed. Usual composition operators, present within the managing program, particularly specify how to add new functionalities, or remove old ones."

When VOs are deployed on physical systems, more virtual nodes might be mapped on the same physical node. This allows for additional peer-to-peer (P2P) communication between the VO's nodes. Some existing or new functionalities could be implemented more effectively by exploiting this direct communication between virtual nodes, mapped on the same physical node.

"I think that one of the most important aspects of our study is how we connect the structure of a system and its functional side," Paduraru told TechXplore. "This can be applied in the future for many categories of problems, both for those that need interconnection of physical agents (e.g. robots that work together to make a path), and software agents (e.g. connecting pieces of software on the cloud)."

In their paper, Paduraru and Stefanescu specifically illustrated their ideas using three examples of VOs for flow management: a tree collector organism, a feeding cell organism and an organism consisting of a collection of connected feeding cell organisms. They then used a simulator for tree collector (TC) organisms to evaluate the benefits of reconfiguration.

Their findings suggest that in dynamically changing environments, reconfigurable structures are more efficient than fixed structures. The researchers also demonstrated how their DSL language, Agapia, could be used to attain fast implementations in VO

simulations.

"We now plan to invest more on support for programming such models, create more test out of the box environments, combine different techniques such as reinforcement learning to create optimization policies without human effort, and finally deploy the virtual organisms in the real world applications," Paduraru said.

More information: Adaptive virtual organisms: A compositional model for complex hardware-software binding. arXiv:1901.04983 [cs.DC]. arxiv.org/abs/1901.04983

AGAPIA v0.1: A programming language for interactive systems and its typing system. DOI: 10.1016/j.entcs.2008.04.087. https://www.sciencedirect.com/science/article/pii/S1571066108003022

APA citation: A new model introduces the concept of adaptive virtual organisms (VOs) (2019, January 31) retrieved 17 September 2019 from https://techxplore.com/news/2019-01-concept-virtual-vos.html