A Networked Robotic System and its Use in an Oil Spill Monitoring Exercise

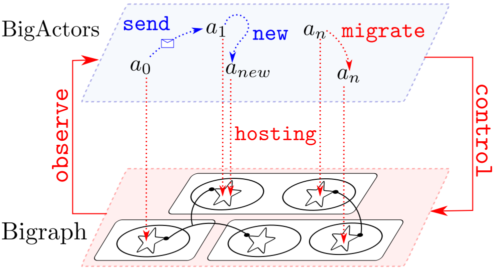
Elói Pereira*†
Co-authors: Pedro Silva†, Clemens Krainer‡, Christoph Kirsch‡, José Morgado†, and Raja Sengupta*
cpcc.berkeley.edu, eloipereira.com
eloi@berkeley.edu

Swarm at the Edge of the Could - ESWeek'13, Montreal, Canada
September 29, 2013

* - Systems Engineering, UC Berkeley
† - Research Center, Portuguese Air Force Academy
‡ - Computer Science Dept., University of Salzburg

Programming the Ubiquitous World

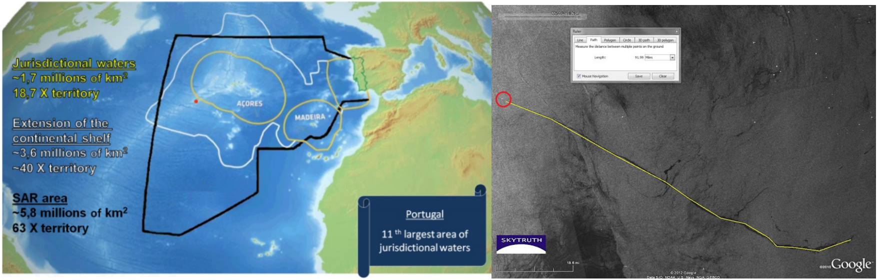
- In networked mobile systems (e.g. teams of robots, smartphones, etc.) the **location** and **connectivity** of “machines” may vary during the execution of its “programs” (computation specifications)
- We investigate models for *bridging* “programs” and “machines” with **dynamic structure** (location and connectivity)
- BigActors [PKSBdS13, PPKS13, PS13] are actors [Agh86] hosted by entities of the physical structure denoted as bigraph nodes [Mil09]



E. Pereira, C. M. Kirsch, R. Sengupta, and J. B. de Sousa, “Bigactors - A Model for Structure-aware Computation,” in ACM/IEEE 4th International Conference on Cyber-Physical Systems, 2013, pp. 199-208.

Case study: Oil spill monitoring scenario

- “Bilge dumping” is an environmental problem of great relevance for countries with large area of jurisdictional waters
- EC created the European Maritime Safety Agency to “...prevent and respond to pollution by ships within the EU”
- How to use networked robotics to monitor and take evidences of “bilge dumping”



Measure the distance between multiple points on the ground

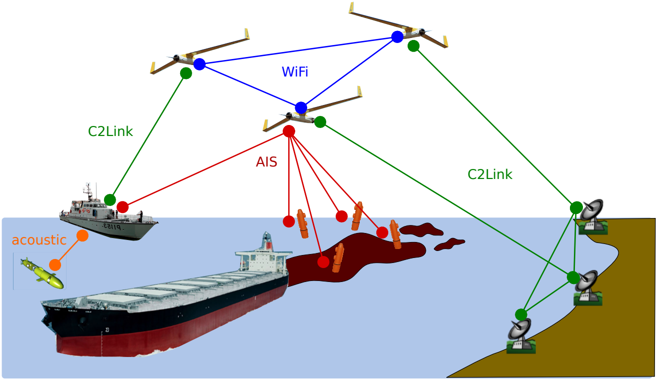
Length: 91.99 Miles

☑ Mouse Navigation

Figure: Portuguese Jurisdiction waters and evidences of “Bilge dumping”.

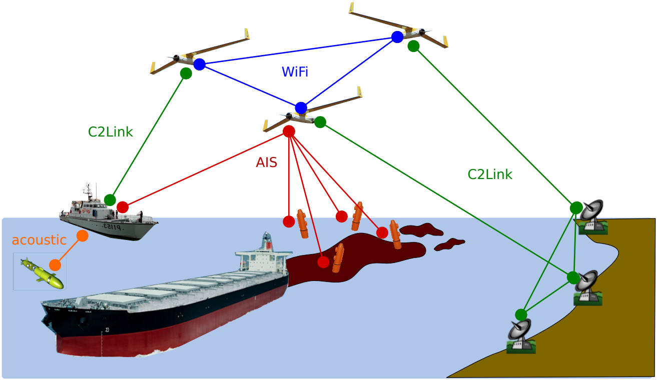
Vehicles and sensors living in the physical world

- Network of vehicles and sensors collaborating to monitor an oil spill caused by a tanker
- UAVs use their optical sensors to detect the oil spill and collects AIS information of vessels and drifters
- A vessel deploys AIS/GPS drifters to forecast the oil spill dynamics
- A submarine collects samples of the oil spill



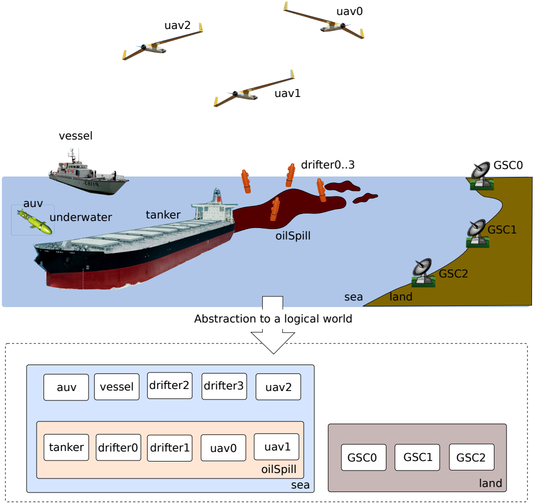
Vehicles and sensors living in the physical world

- How to model vehicles and sensors embedded in the physical world in a logical abstraction?
- How to program computing entities living and interacting in this logical world?

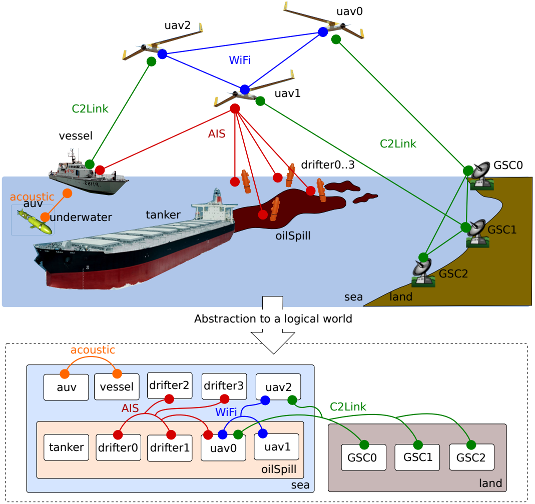


E. Pereira, C. Potiron, C. M. Kirsch, and R. Sengupta, "Modeling and controlling the structure of heterogeneous mobile robotic systems: A bigactor approach," in 2013 IEEE International Systems Conference (SysCon), 2013, pp. 442-447.

Nesting locations as a forest



Connectivity as an hypergraph



Programming the logical world

- BigActor program that keeps sending the UAV to the oil spill location:

  ```
  "gotoOil" hosted_at "uav0" with_behavior{
    loop{
        MOVE_HOST_TO oilSpill
    }
  }
  ```

- This program would be interpreted to a series of physical UAV commands:

  ```
  autopilotWaypointCtr(Lat0,Long0,Alt0,radius)
  autopilotWaypointCtr(Lat1,Long1,Alt1,radius)
  autopilotWaypointCtr(Lat2,Long2,Alt2,radius)
  autopilotWaypointCtr(Lat3,Long3,Alt3,radius)
  ...
  ```

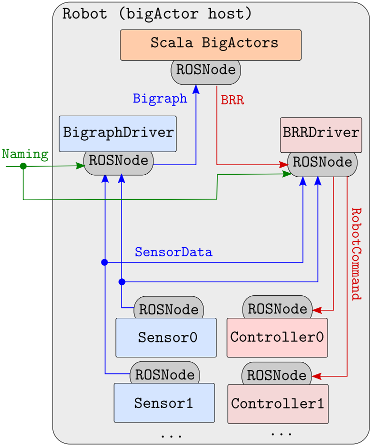
- Logical space programming focuses on “what” you want to do rather than “how” to do it

Video: Networked vehicles and sensors demonstration

FIDELIDADE
Oil Spill Monitoring using Networked Vehicles and Sensors
REP13 Exercise, Portimão, Portugal
July 2013

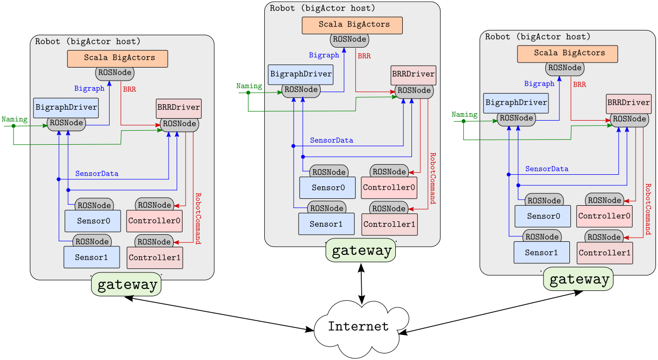
BigActor Runtime Environment for Networked Robotics

- BigActor implementation for networked robotics is comprised of two major software components:
    - BigActor Domain Specific Language using Scala Actors
    - BigActor Runtime Environment implemented over the Robot Operating System (ROS) - a widely used middleware for robotics



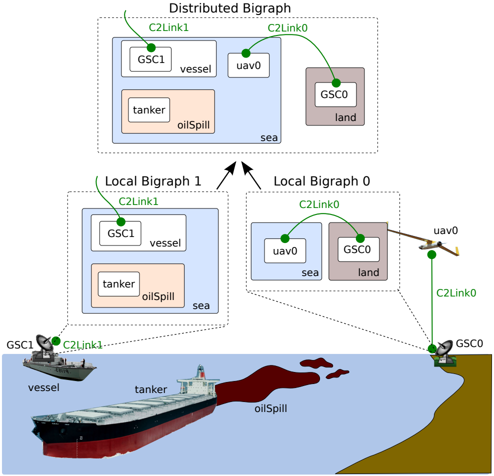
Distributed Bigraph Estimation

- Each robot observes the world locally as a bigraph
- Local observations are shared amongst robots over the internet
- Each robot calculates a bigraph estimate of the overall system
- When a robot receives a distributed bigraph estimate (`DBigraph`), it fuses it with its own estimate, using a bigraph fuser:
  `bigraphFuser:: (DBigraph,DBigraph) => DBigraph`



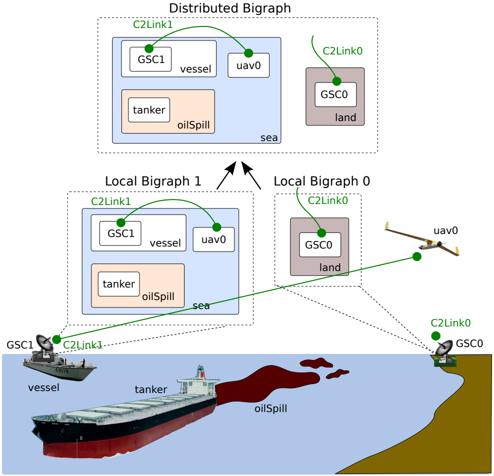
Example: handover of UAV control authority

```
"handover" hosted_at "GCS0" with_behavior{
   HANDOVER uav0 TO GCS1
}
```



Example: handover of UAV control authority

```
"handover" hosted_at "GCS0" with_behavior{
   HANDOVER uav0 TO GCS1
}
```



Programming models with space as first-class citizen

- Ambient Calculus models bounded locations where computation may occur [CG98]
- Borcea et al. introduce Spatial Programming (SP) - a space-aware programming model for outdoor distributed computing. In SP, physical spaces are abstracted as circular regions that circumscribes the physical space [BIK$^+$04]
- Gaia [RC00] is a software infrastructure that supports *Active Spaces* - a model that maps the abstract perception of space into a first-class software citizen. Active spaces store location information and define their behaviour
- Bigraphs [Mil09] provide a rich model that entails both location and connectivity.
- In contrast with SP and Active Spaces, Bigraph provides a nested location model.
- Bigraphs do not explicitly represent intersection of spaces (Shared Bigraphs introduce this concept).

Comparison with other spatial programming models

- BigActors combine a widely known model of concurrency combined with a bigraph abstraction of the world
- SP is based on *Smart Messages* (SMs) model of concurrency while the BigActor model uses Actors.
- SM model is based on migration of computing units between entities a shared memory infrastructure
- Actors are based on asynchronous message-passing
- In the Gaia framework, Active Spaces encapsulate themselves state and behaviour of the physical spaces
- The BigActor Model specifies the physical space in the bigraph and the computing entities as actors. This provides a separation of concerns between the physical space and what actually changes it.

Conclusions

- Robots live in a physical world
- We explore BigActors as a model for logically program networked robotics
- Programming at a logical level reduces operational complexity
    - spatial programming example
    - hand-over manoeuvre example
- We introduce a BigActor Runtime Environment that runs over ROS and Scala Actors
- The model and the implementation was successfully demonstrated over the summer specifying missions of networked vehicles performing an environmental monitoring scenario.

"*We can only see a short distance ahead, but we can see plenty there that needs to be done.*" Alan Turing [Tur50]

"*I was concerned... to ask what the limits of computation may be. This interaction business began to seem to me to be breaking the mould...*" Robin Milner [Berger interview]

Thank You!

References I

Gul Agha.
Actors: a model of concurrent computation in distributed systems.
MIT Press, Cambridge, MA, USA, 1986.
ISBN 0-262-01092-5.

L. Birkedal, S. Debois, E. Elsborg, T. Hildebrandt, and H. Niss.
Bigraphical models of context-aware systems.
In *Foundations of software science and computation structures*, pages 187–201. Springer, 2006.

P. Bonnet, J. Gehrke, and P. Seshadri.
Querying the physical world.
IEEE Personal Communications, 7(5):10–15, 2000.
ISSN 10709916.

C. Borcea, C. Intanagonwiwat, P. Kang, U. Kremer, and L. Iftode.
Spatial programming using smart messages: design and implementation.
In *24th International Conference on Distributed Computing Systems, 2004. Proceedings.*, pages 690–699. Ieee, 2004.
ISBN 0-7695-2086-3.

L. Cardelli and A. Gordon.
Mobile ambients.
In *Foundations of Software Science and Computation Structures*, pages 140–155. Springer, 1998.

References II

S. Debois.
Computation in the informatic jungle.
To appear. Draft available at
http://www.itu.dk/people/debois/pubs/computation.pdf, 2010.

R. Milner.
Communicating and mobile systems: the π-calculus.
Cambridge Univ Press, 1999.

Robin Milner.
The Space and Motion of Communicating Agents.
Cambridge University Press, 2009.
ISBN 978-0-521-73833-0.
I-XXI, 1-191 pp.

G. Perrone, S. Debois, and T.T. Hildebrandt.
A model checker for bigraphs.
In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, pages 1320–1325. ACM, 2012.

E. Pereira, C. Kirsch, R. Sengupta, and J. Borges de Sousa.
Bigactors - a model for structure-aware computation.
In *to appear at the 4th International Conference on Cyber-Physical Systems*. ACM/IEEE, April 2013.

E. Pereira, C. Potiron, C. Kirsch, and R. Sengupta.
Modeling and controlling the structure of heterogeneous mobile robotic systems: A bigactor approach.
In *to appear at the IEEE Systems Conference*. IEEE, April 2013.

References III

Eloi Pereira and Raja Sengupta.
Computation Over Worlds With Dynamic Structure.
In *III International workshop on Bigraphs*, 2013.

Manuel Roman and RH Campbell.
Gaia: enabling active spaces.
Proceedings of the 9th workshop on ACM . . . , pages 229–234, 2000.

DA Randell, Z Cui, and AG Cohn.
A spatial logic based on regions and connection.
KR, 1992.

A. Turing.
Computing Machinery and Intelligence.
Mind - A quarterly review, 59(236):433–460, 1950.

Leslie G Valiant.
A bridging model for parallel computation.
Communications of the ACM, 33(8):103–111, 1990.

John von Neumann.
First draft of a report on the EDVAC.
Technical report, University of Pennsylvania, June 30, 1945.