In objectland...
There are many ways to combine objects.
But sometimes, it's like Abbott's flatland...
We just miss a dimension or two...
We would like to get up and walk, to...
...to run and escape from object-land!
Fail.
Of course
If you ignore half of it, these are the feet...
Assume this would be up here...
Let's pretend here would be a gap...
...and here.

Imagine there would be a leg...

We can try to fix it with objects...
But, why not
Compose it if from smaller part?

[Ossher 2007]
The same small parts...
...from which objects are built?
In fragmentland...
...objects are not monolithic.
First, separate head from body.
Allows us to inject new behavior…

[Haupt and Schippers 2007]
Allows us to keep track of references…

[Lienhard et al. 2008]
Cell Types and Notation

- **H1**: Head or Alias cell
- **@name**: Slot cell
- **#name**: Function cell
- **int str a[]**: Primitive cell
- *****: Custom lookup cell
- **B**: Branching cell
- **Message cell**: Message cell
- **Reply cell**: Reply cell
• Each cell consists of:
  – A local lookup function
  – A pointer to next cell
  – An optional payload
• Lookup works as follows:
  – Delegation based, message based
  – If local lookup fails, then delegate to next cell

The CELL Model in a Nutshell
CELLTALK

Smalltalk made of Cells
CELLTALK + Traits
CELLTALK + Aliases

Changes to CELLTALK
- Instance creation
- Method install

p = Point.send("x:y:", 2, 3)
q = Point.send("x:y:", 4, 5)
p.send("x=", q.send("x"))
• List of cell types
  – Limited or extensible?
• How to model supersends?
  – A resend is not a supersend
• Performance?
  – Current prototypes in Java and Python
• Formalization

Open Issues and Questions