LIANE—Composition for Active Networks

Samrat Bhattacharjee **Ken Calvert** **Ellen Zegura**

Networking and Telecommunications Group
College of Computing
Georgia Institute of Technology
Atlanta, Georgia, USA

`http://www.cc.gatech.edu/projects/canes`

Sponsors: DARPA, NSF

Problem Statement

Dynamically create per-user network services whose behavior can be analyzed with respect to safety of the active network and progress of the composite computation.

- Users can create dynamic services *and*
- Network is protected

$\Rightarrow$ Tension between network safety and programmability

- Create dynamic services by *composition* and
- Analyze safety by *formal methods*

Approaches towards Composition

- Turing-complete language
- Object-oriented composition
- Software bus — POLYLITH
- Event-driven framework — *micro-protocols* in x-kernel

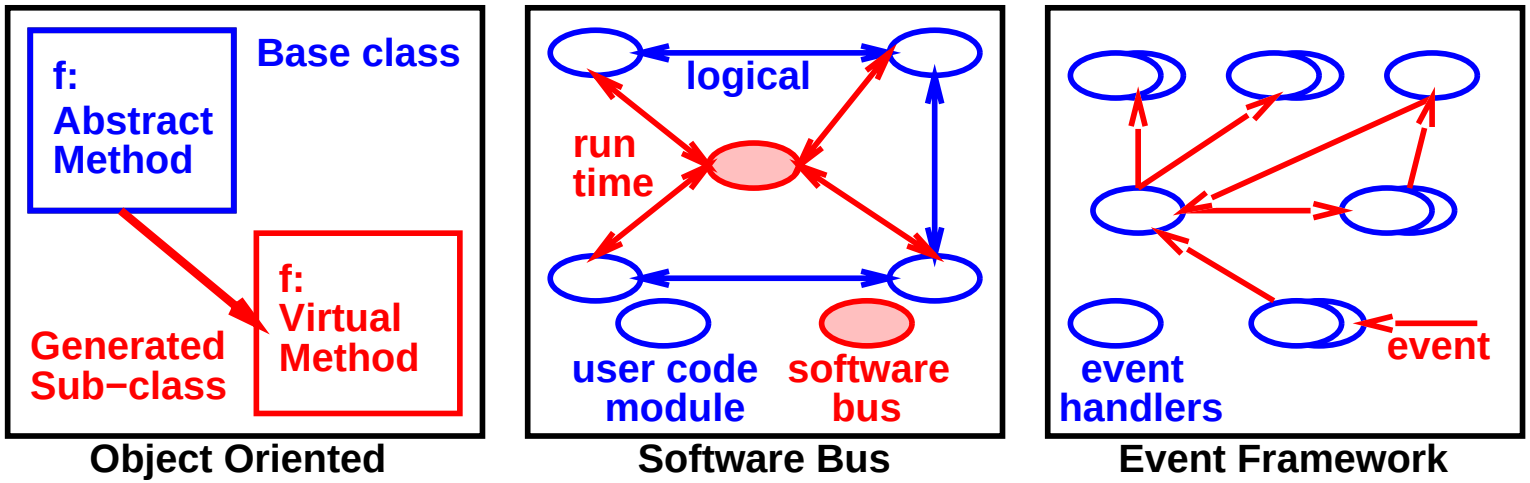


Figure 1: Composition Mechanisms

None are Perfect!

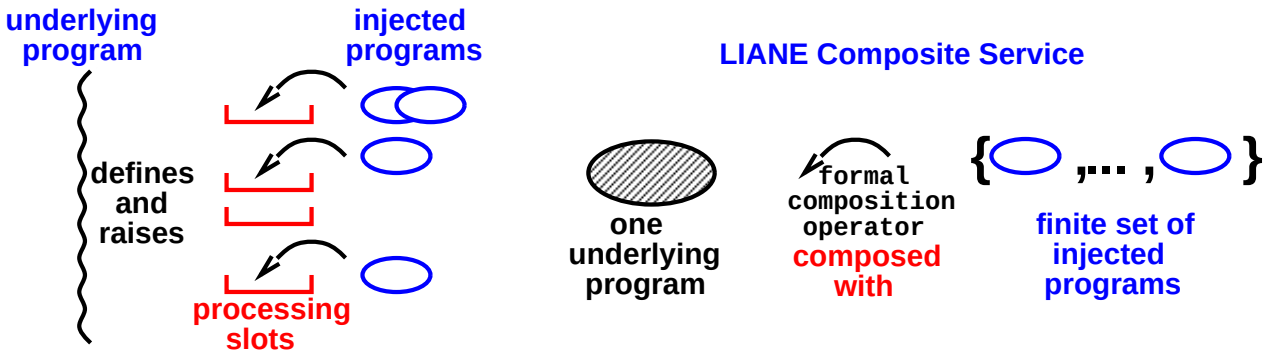
- Application structure is fixed — Object-oriented, POLYLITH
- Variable function signature — POLYLITH, *micro-protocols*
- Formal properties lay entirely on provable properties of code modules — in *all* cases

Need *active* software interconnect

- implements run-time composition framework
- can be used to assert provable node properties

LIANE — A formal composition framework

- Underlying program — supplied by node provider
- Processing slots — localizes user code
- Injected programs — selected, provided by users



- Restrictions and obligations of injected code
- Formal transformation technique to form composite program

LIANE — Example

Parse packet, obtain source s, destination d
⟨**Slot 0:[null]**⟩ {trace route, caching, select route table, discard}

Outputlist := ()
i := **Lookup**(d, `route table` R)
if Outputlist = () then ⟨**Slot 1:[null]**⟩ {error messages to source}
⟨**Slot 2:[null]**⟩ {send route back, select alternate interface}

if i is congested then ⟨**Slot 3:[discard]**⟩ {cong. control algorithm}
else ⟨**Slot 4:[null]**⟩ {scheduling algorithm}
enqueue packet for i.

Figure 2: Example Underlying Program with processing slots

Status

- Used to implement AN-SIM
- AN-SIM nodes simulate composition using LIANE
- Ongoing work on UNITY model