Reasoning about concurrency with separation logic

Kamal Lodaya
(joint work with Kalpesh Kapoor, IIT Guwahati, and Uday Reddy,
University of Birmingham)

How do you reason about concurrent programs which use shared storage? There are a few ideas so far: Ashcroft (1975) suggested using global invariants to express unchanging properties of the shared store.
Owicki and Gries (1976) suggested checking that assertions made in one process do not interfere with those made in a parallel one. Jones (1983) suggested using assertions which a process could *rely* on from other processes, and assertions which it could *guarantee* for other processes, an idea similar to earlier work of Misra and Chandy (1981). After the development of separation logic (see Reynolds, 2000), O'Hearn (see 2007) introduced its use for concurrent programs, with different processes *owning* different parts of store (ownership could change dynamically).
Bornat, Calcagno, O'Hearn and Parkinson (2005) refined this idea using a notion of processes having *permissions* to access the shared store.
In 2006, we combined permissions with Ashcroft's global invariants.
Vafeiadis and Parkinson (2007) combined separation logic with Jones's rely-guarantee technique. This talk surveys these ideas.