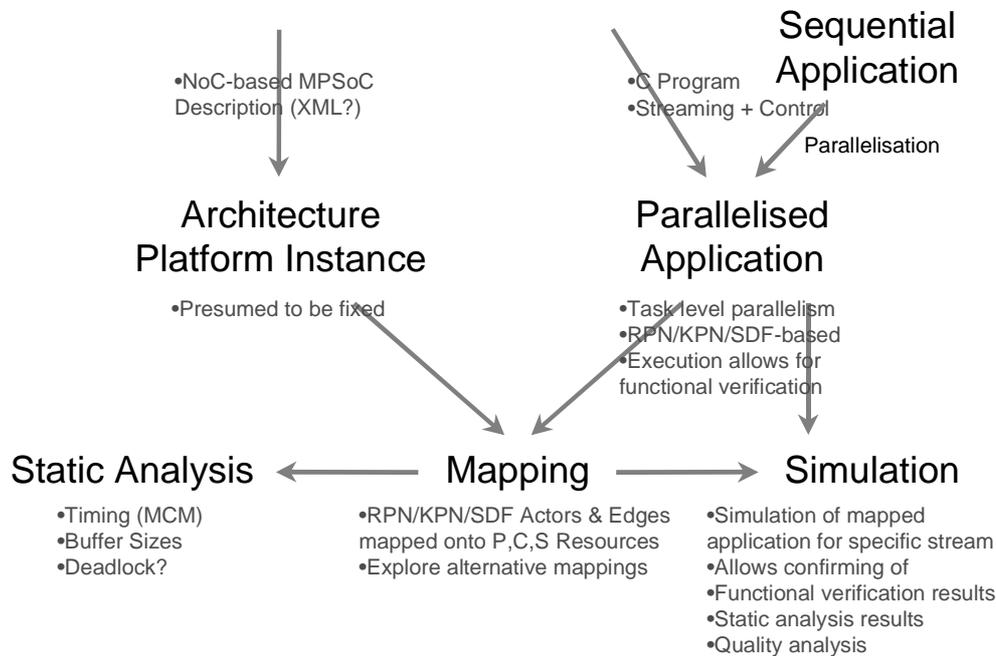


## Design flow



### Remarks:

- For development of new applications, we prefer to program directly in a parallel (Yapi) specification.
- Preferably no extensive iterations involving simulation. For design-space exploration (DSE) we can iterate over mapping and static analysis.
- How can we integrate Design Space Exploration for alternative Architecture Platform Instances?
- Do we really need both Static Analysis and Simulation?

**Terminology:** system (product), application, job, task. An application is started by the user. A job is started by an application. For streaming, a job corresponds to an SDF graph, and a task to an actor.

An actor should have a sharp bound on its execution time (ET). If the ET is very different for different input data, we prefer to describe that using scenarios.

We would like to reason (about timing) already at a high level. In order to do that, timing information at the level of actors is required, and (partial) mapping decisions and some architecture information is needed to provide timing information.

Is Poosl suitable for high-level reasoning?

### Poosl vs. MCM analysis:

MCM analysis is restricted to a class of streaming that can be described in SDF, thereby allowing efficient analysis of timing and buffer requirements.

Poosl describes a much broader class that includes non-deterministic behaviour.

As a result, the state-space is very large and most of the involved mathematical problems are intractable.

Furthermore, currently there is no tooling in Poosl to solve those problems.

Therefore simulation is the current practice with Poosl. The reliability and applicability of Poosl results depends largely on an accurate statistical modelling of the environment (the input data).

This suggests that static MCM analysis is suitable for streaming. For dynamic and non-deterministic behaviour and for soft real-time situations, a simulation-based (Poosl) approach may be more interesting. Can we use this approach to derive some metrics concerning perceptive quality ?