Centrum voor Wiskunde en Informatica

Distributed Dataspace Architectures

Projects + partners:

STW: EIF.3959 (KUN)
Jozef Hooman (project leader), Ulrich Hannemann

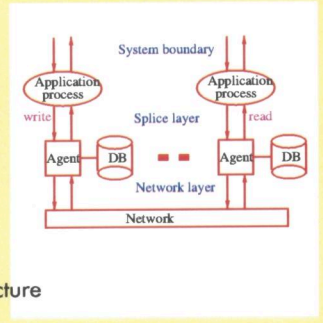
Progress: CES.5009 (CWI)
Jaco van de Pol (project leader), Simona Orzan, Miguel Valero Espada

Industrial partner: Thales Nederland.

Two instances of distributed dataspace architectures:

SPLICE

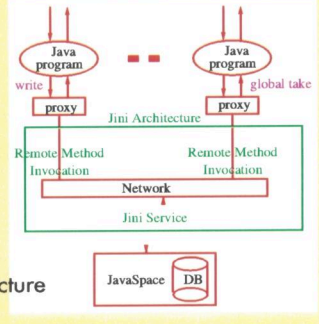
. publish/subscribe paradigm
. records with keys, sorts
. data is equipped with time stamps
. developed by Hollandse Signaal-apparaten bv. (now Thales)



An overview of the Splice architecture

JAVASPACES

. transaction mechanism
. leasing of data space entries
. notification
. developed by SUN on top of JINI



An overview of the JavaSpaces architecture

Comparison:

	Splice	**Javaspaces**
application:	fast data distribution	distributed algorithms
data storage:	distributed	centralized
organization:	database with key fields	multiset
primitives:	local read, write, take	global read, write, take
data removal:	overwrite mechanism	leasing
data selection:	queries	template matching

Research goals:

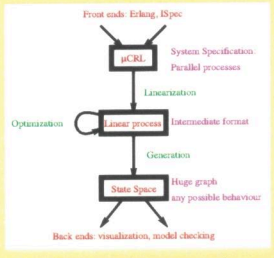
Goal is the analysis of dataspace systems. The starting point is developing and validating formal models of Splice and JavaSpaces. Next applications on top of these architectures are specified. Analysis includes formal specification and verification of:

- design steps for applications - fault tolerance (self healing) - real-time behaviour and performance - transparent replication

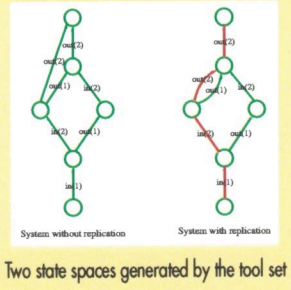
Complementary Formal Methods:

µCRL

. system is modelled as a number of parallel processes
. based on process algebra and algebraic data types
. state space can be generated automatically
. debugging by simulation, vizualization
. verification by equivalence checking and model checking



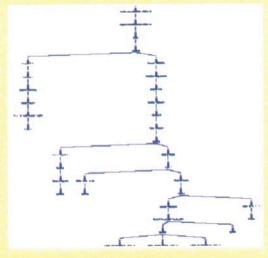
An overview of the µCRL tool set



Two state spaces generated by the tool set

PVS

. interactive theorem prover
. based on strongly-typed higher-order logic.
. hierarchy of theories, containing definitions and theorems.
. PVS has many state-of-the-art decision procedures.



A proof constructed with the PVS theorem prover

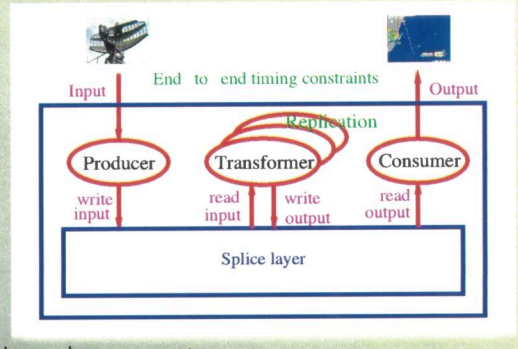
Comparison:

µCRL tool set	**PVS theorem prover**
operational model (simulation)	denotational semantics (reasoning)
automatic verification (also counter examples)	interactive verification (rerunnable proofs)
rapid prototyping (investigate design space)	step-wise refinement (top-down verification)
concrete instances (e.g. 6 inputs, 3 transformers)	general systems (m inputs, n transformers)
equality between processes (e.g. trace equivalence)	satisfies requirements (safe output)

Results:

. Formal models of Splice and JavaSpaces in µCRL and PVS, apt for simulation and reasoning
. Expressivity results on basic primitives of Splice
. Proposal for timed extension of Splice primitives
. Formal verification of a number of applications:

Applications:
- coordination of parallel workers in JavaSpaces.
- real-time behaviour of radar-display system on Splice
- transparent replication on Splice



The radar-display application
Note the replication and the end-to-end time constraints

www.cwi.nl