Very Lightweight Agents for Reactive, Fault-Tolerant Real-Time Systems

Jae C. Oh, Madhura S. Tamhankar, and Daniel Mossé

Department of Electrical Engineering and Computer Science

Syracuse University

Department of Computer Science

The University of Pittsburgh

Keywords for Motivation

- Large-scale

Keywords for Motivation

- Large-scale
- Real-time

Keywords for Motivation

- Large-scale
- Real-time
- High-speed data aquistition

Keywords for Motivation

- Large-scale
- Real-time
- High-speed data aquistition
- Fault-tolerant

Keywords for Motivation

- Large-scale
- Real-time
- High-speed data aquistition
- Fault-tolerant
- Adaptive

Keywords for Approach and Goals

- Multiagent

Keywords for Approach and Goals

- Multiagent
- Very Lightweight

Keywords for Approach and Goals

- Multiagent
- Very Lightweight
- Fast

Keywords for Approach and Goals

- Multiagent
- Very Lightweight
- Fast
- Small footprint

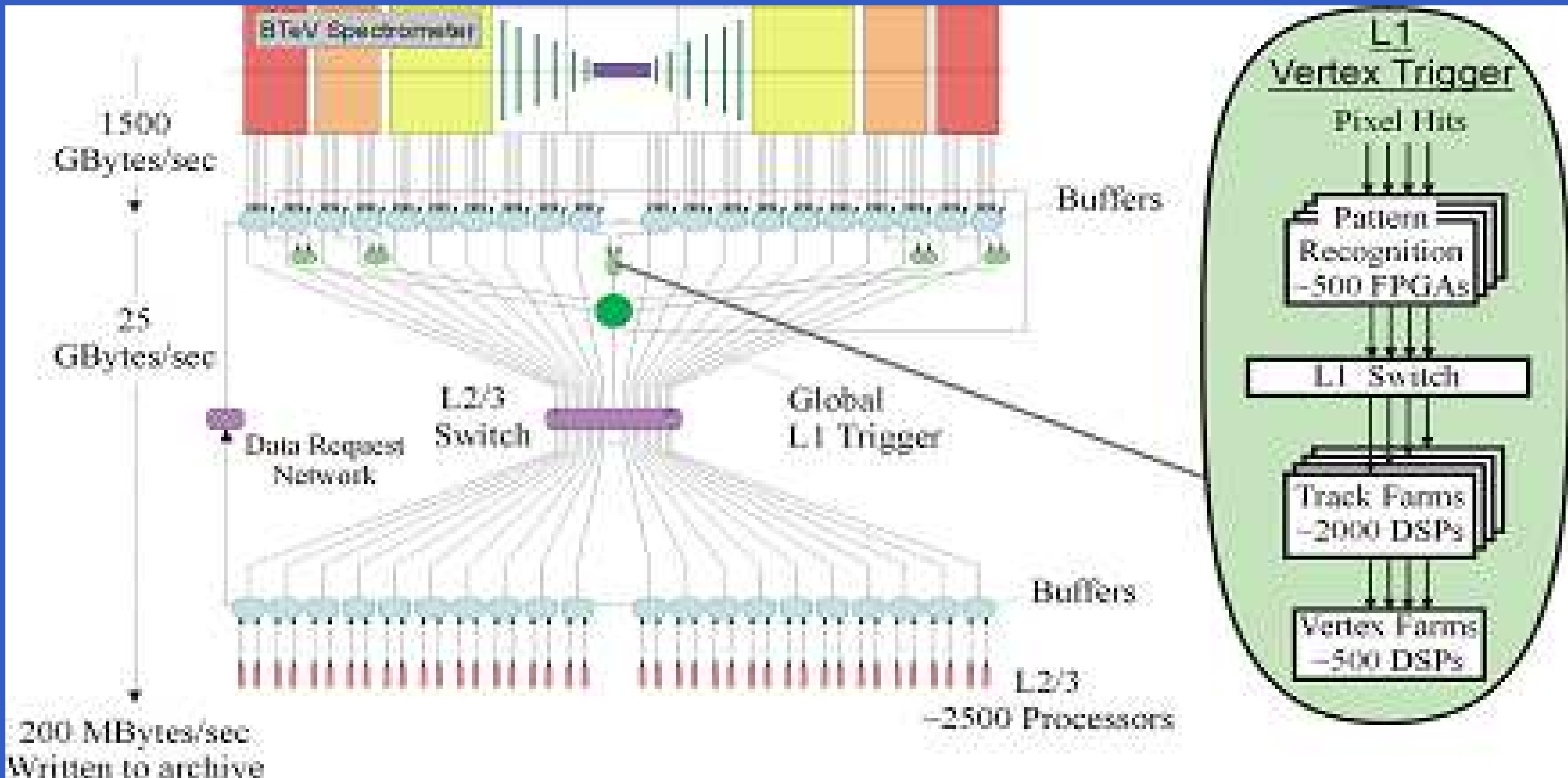
Keywords for Approach and Goals

- Multiagent
- Very Lightweight
- Fast
- Small footprint
- Adaptive

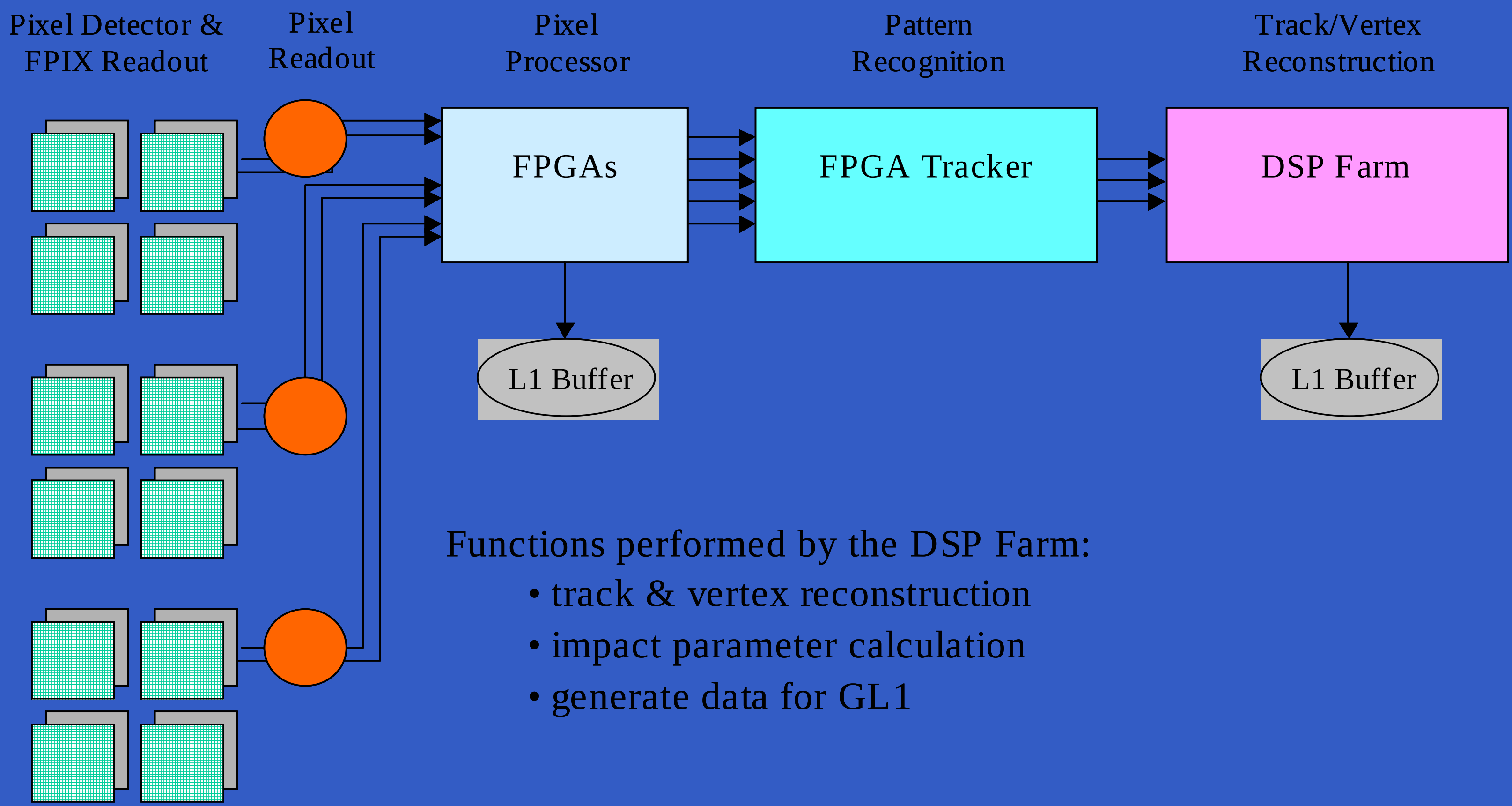
Keywords for Approach and Goals

- Multiagent
- Very Lightweight
- Fast
- Small footprint
- Adaptive
- Intelligent

BTeV triggering and data acquisition s



Level 1 Pixel Trigger



Functions performed by the DSP Farm:

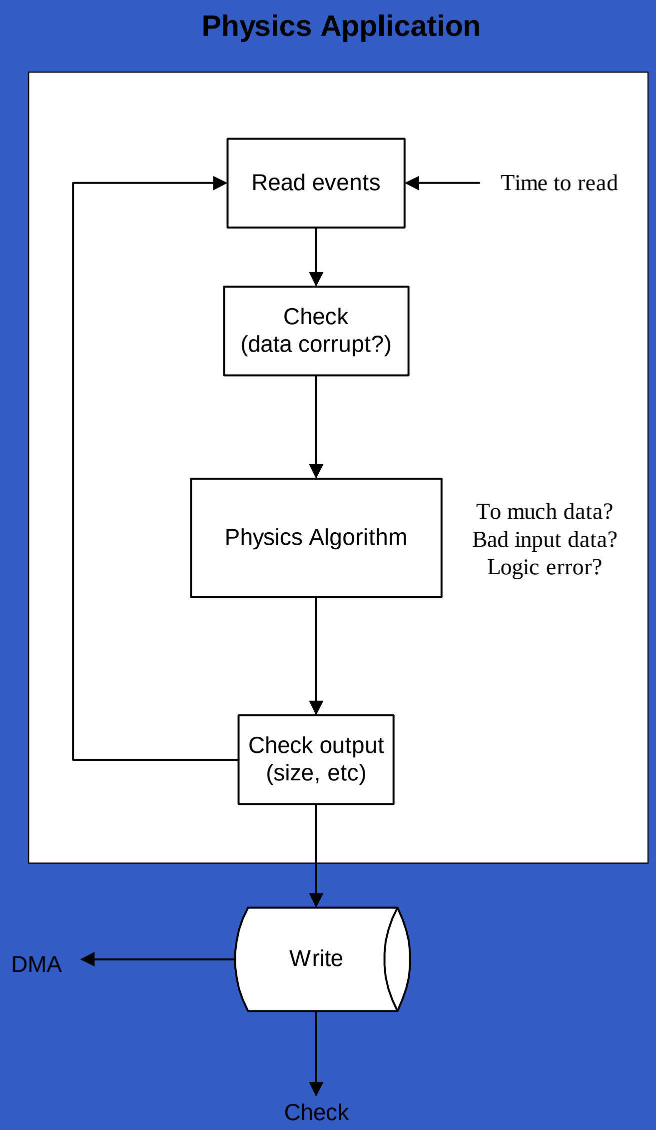
- track & vertex reconstruction
- impact parameter calculation
- generate data for GL1

Figure 0: Adopted from BTeV-doc-308-v1 (Gottschalk)

BTeV triggering and data acquisition s

- About 2,500 DSPs (L1)
  - Fast data aquisition
  - Fault monitoring
  - Fault-tolerant
  - Adaptive to failure
  - Run physics applications

Event-loop of a physics application



Very Lightweight Agents

- Monitor hardware integrity

Very Lightweight Agents

- Monitor hardware integrity
- Monitor software integrity

Very Lightweight Agents

- Monitor hardware integrity
- Monitor software integrity
- Intelligent and Adaptive (e.g., error prediction, correction)

Very Lightweight Agents

- Monitor hardware integrity
- Monitor software integrity
- Intelligent and Adaptive (e.g., error prediction, correction)
- Cooperative

Very Lightweight Agents

- Monitor hardware integrity
- Monitor software integrity
- Intelligent and Adaptive (e.g., error prediction, correction)
- Cooperative
- Reactive

Very Lightweight Agents

- Monitor hardware integrity
- Monitor software integrity
- Intelligent and Adaptive (e.g., error prediction, correction)
- Cooperative
- Reactive
- Proactive

Very Lightweight Agents

- Monitor hardware integrity
- Monitor software integrity
- Intelligent and Adaptive (e.g., error prediction, correction)
- Cooperative
- Reactive
- Proactive
- Fast

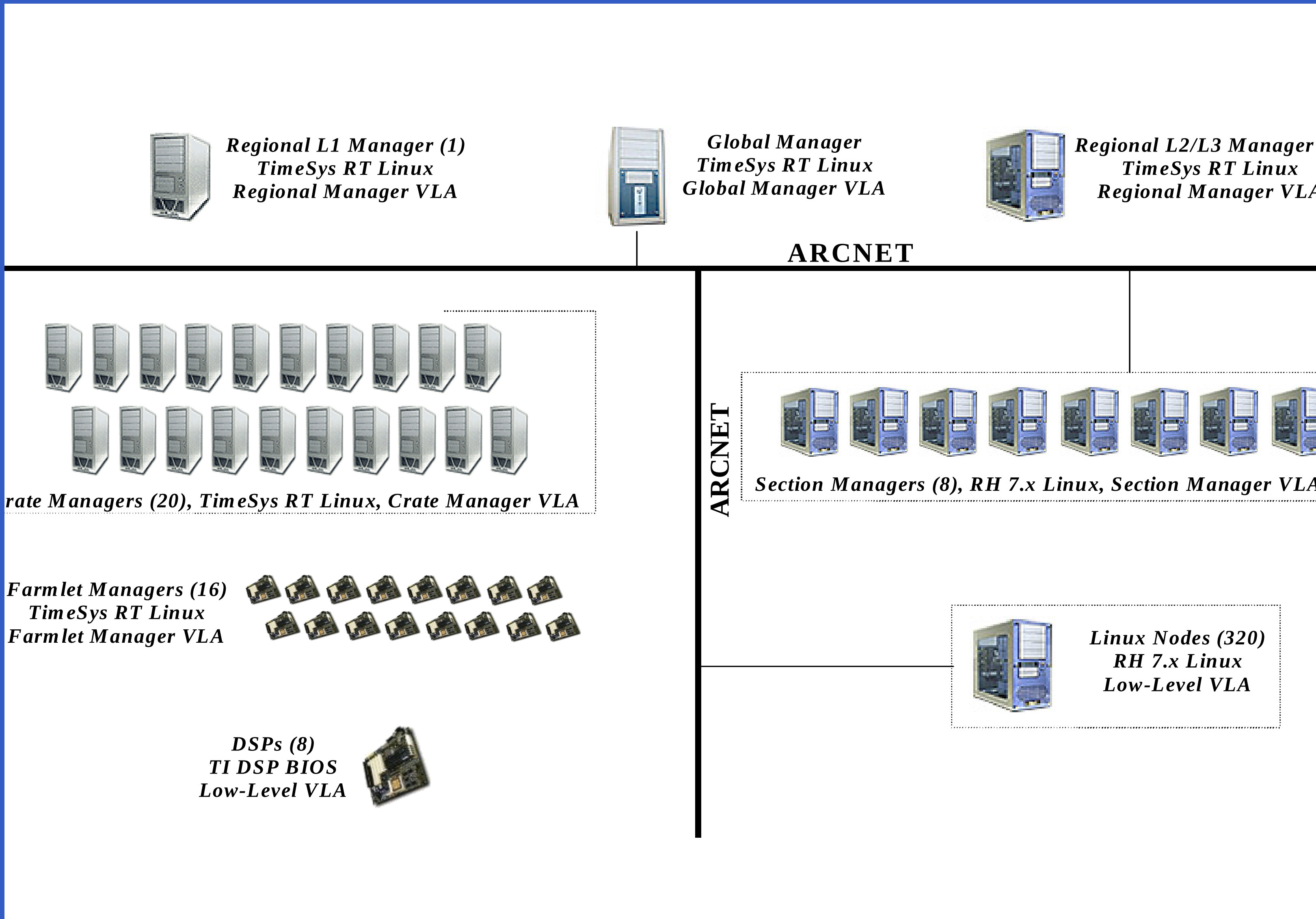
Very Lightweight Agents

- Monitor hardware integrity
- Monitor software integrity
- Intelligent and Adaptive (e.g., error prediction, correction)
- Cooperative
- Reactive
- Proactive
- Fast
- Small footprint

Very Lightweight Agents

- Monitor hardware integrity
- Monitor software integrity
- Intelligent and Adaptive (e.g., error prediction, correction)
- Cooperative
- Reactive
- Proactive
- Fast
- Small footprint
- Hierarchical

Envisioned VLA hierarchy (Prelimiar



VLA design theoretical background

- Brooks' subsumption architecture
- designed for robot controls
- Small reactive Augumented Finite State Machines – no full-blown explict k nowledge representaion
- Higer level "subsumes" lower level control when appropriate

VLA theoretical background (cont.)

- Intelligence “emerges” when simple components interact
- BTeV/RTES deals with real hardware components (robot control systems too)
- Realtime managements, Fault-tolerant, small footprint, fast, adaptability

Rule subsumption

```
Function action (p:  P): A
var fired:  P(R)
var selected :  A
begin
   fired := {(c,a)|(c,a) ∈ R and p ∈ c }
   for each (c,a) ∈ fired do
      if ¬(∃(c′,a′) ∈ fired such that (c′,a′) ≺ (c,a) then
         return a
      end-if
   end-for
   return null
end function action
```

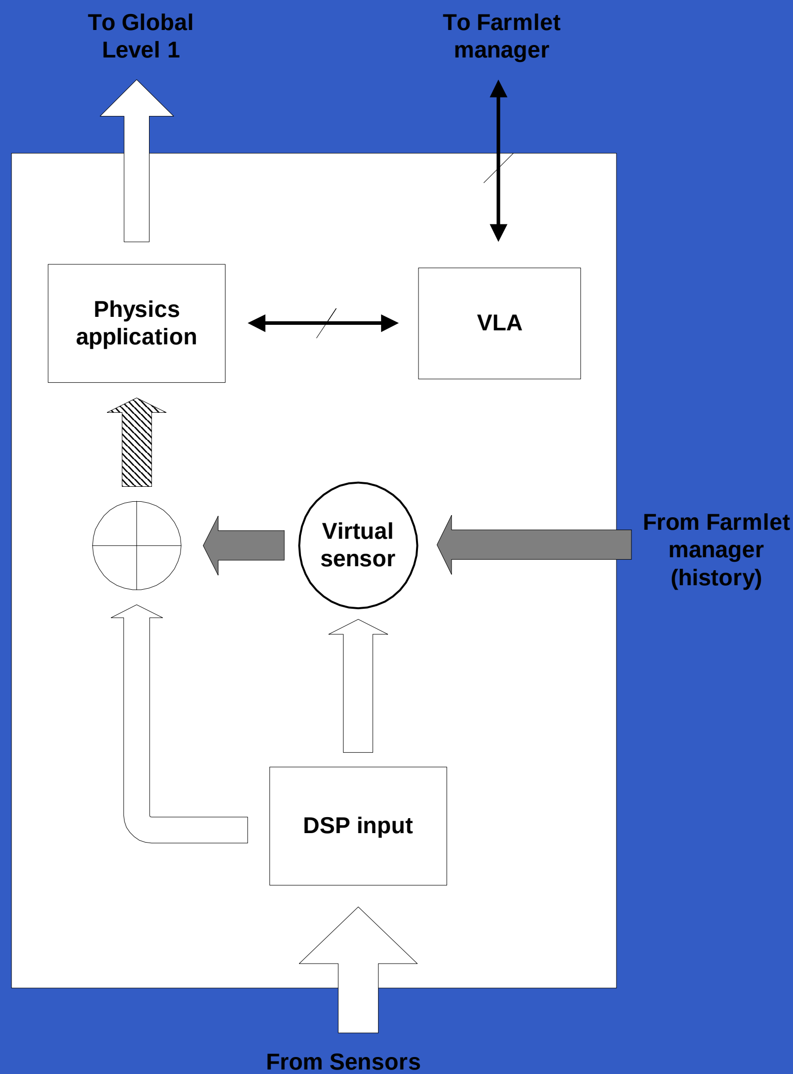
Suggested Modifications to the SA

- The division between layers is too rigid: let layers to be able to communicate if needed while keeping the communication traffic minimal.
- Allow blackboard architecture: VLAs can communicate via this facility.
- Introduce proactivness in the subsumption architecture.
- Inter-VLA subsumption and Intra-VLA subsumption: hierarchy of VLAs

The notion of virtual sensors

- When a hardware sensor fails, a virtual sensor can act
- Temporary feed from history (extraploation)
- Temporary feed from near by sensors
- Usefulness in the RTES environment – yet to be determined

Virtual sensor with VLA



Subsumption Rules for Farmlet VLAs

FR1 Condition: no communication with a L1-level VLA for a certain period of time. Possible causes: The DSP may be down; Com-link may be down; The DSP may be overloaded. Actions: Ping the DSP; reset the DSP; reboot the DSP.

FR2 Condition: I/O error or File I/O error when accessing a history file. Possible causes: File doesn't exist; Disk full Invalid file pointer; Communication problem with a file server. Actions: Notify higher level entities including human operator.

FR3 Condition: the first start-up after a power loss. Actions: Perform data and hardware integrity tests.

FR4 Condition: Received data integrity problem notification from a lower level VLA. Actions: Find the hardware that causes problem; Activate the virtual sensor and channel history data to the virtual sensor, if appropriate.

FR5 Condition: The Farmlet CPU utilization drops below a certain level. Action: Run additional diagnostic procedure, communicate with other VLAs, perform preventive actions, and check not-so-critical conditions.

FR6 Condition: Timer interrupt for periodic checking is received. Action: Perform the required procedure.

Priority $FR3 \prec FR1 \prec FR4 \prec FR2 \prec FR6 \prec FR5$.

Rules for L1 level VLAs

- L1R1 Condition: Event data corrupted. Action: report to the Farmlet manager. Check hardware.
- L1R2 Condition: Too much event data. Actions: report to Farmlet manager.
- L1R3 Condition: Too little data. Actions: report to Farmlet manager. Note that the Farmlet manager may activate corresponding virtual sensor.
- L1R4 Condition: No input from sensors Actions: report to the Farmlet manager. Test hardware integrity.
- L1R5 Condition: Overloaded DSP. Actions: report to the Farmlet manager.
- L1R6 Condition: Timer interrupt for periodic testing task has been received. Action: Perform the test.
- L1R7 Condition: Certain additional CPU cycles can be spared for the VLA. Action: Run additional diagnostic procedure, communicate with other VLAs, perform preventive actions, check not-so-critical conditions, etc.

Priority $L1R4 \prec L1R5 \prec L1R1 \prec (L1R3 = L1R2) \prec L1R6 \prec L1R7$.


L1R4 : No input from sensors (notified from the physics application. The physics application may not have been able to do anything)--> Report to the Farmlet manager. Test hardware integrity
L1R5: Overloaded DSP --> Report to the Farmlet manager
L1R1 : Event data are corrupted (notified from the physics application) --> Report to the Farmlet manager. Investigate if a hardware problem
L1R3 : Too little data. (notified from the physics application after attempting to extrapolate) --> Report to Farmlet manager
L1R2 : Too much event data (notified from the physics application after throwing away certain percentage of the data)--> Report to Farmlet manager
L1R6 : Timer interrupt for periodic testing task has been received --> Perform the test
L1R7 : Additional CPU cycles can be spared for the VLA --> Run additional diagnostic procedure, check not-so-critical conditions

VLA Prototypes

- Several prototypes built for an experimental DSP board
- Two TI DSPs on board
- Need more DSPs to study interactions among VLAs
- Currently, building VLAs for a DSP board with 8 DSPs.
- Please visit the poster presentation session for details of prototypes

Conclusion

- The BTeV environment calls for fast, fault tolerant, adaptive agents
- Concept of VLA is being investigated
- Based on the subsumption architecture
- Several pre-prototypes have been built (limited rules and actions)