

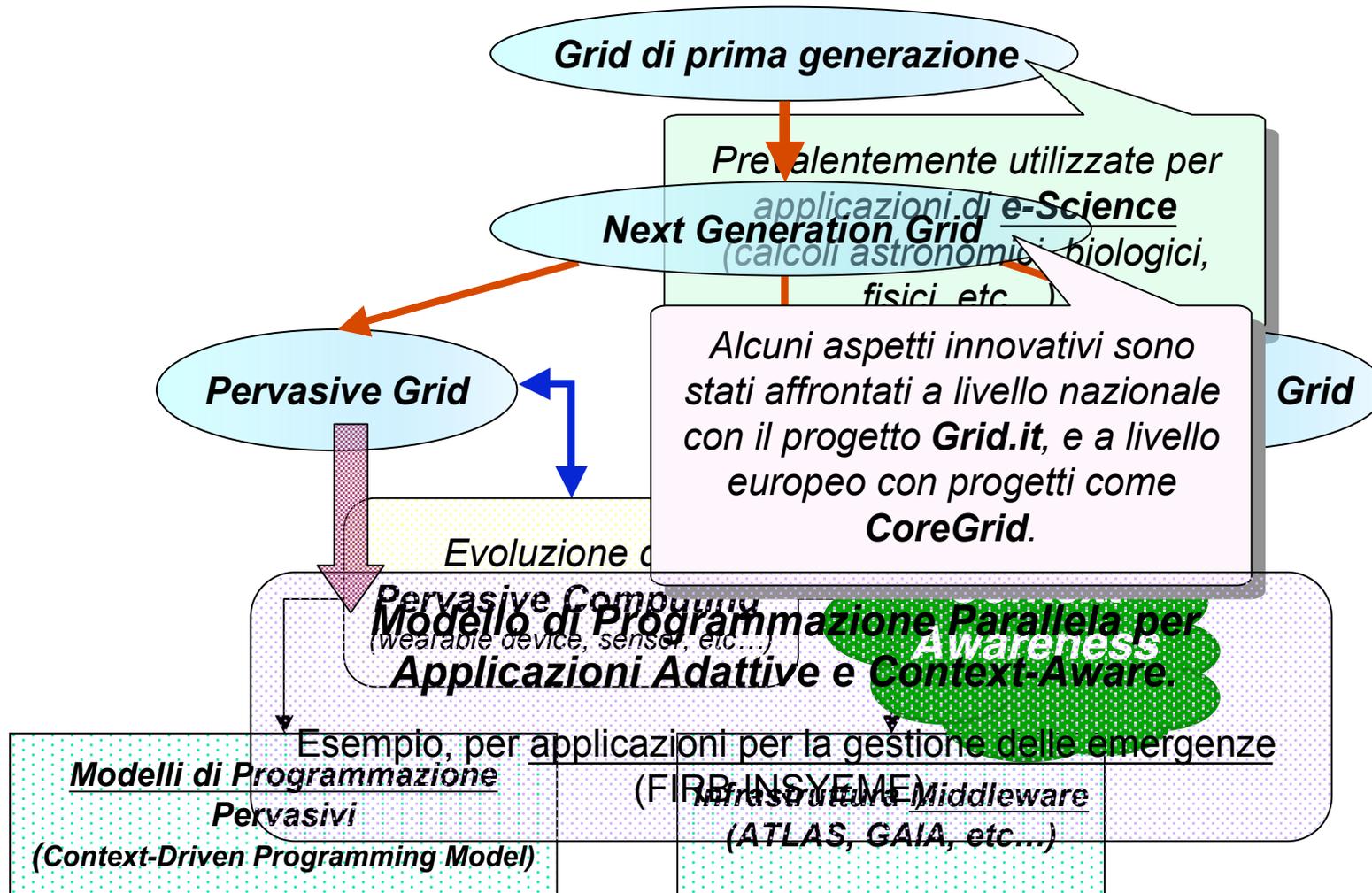
Seminario su InSyEme

Progetto FIRB In.Sy.Eme
(Integrated System for Emergency)

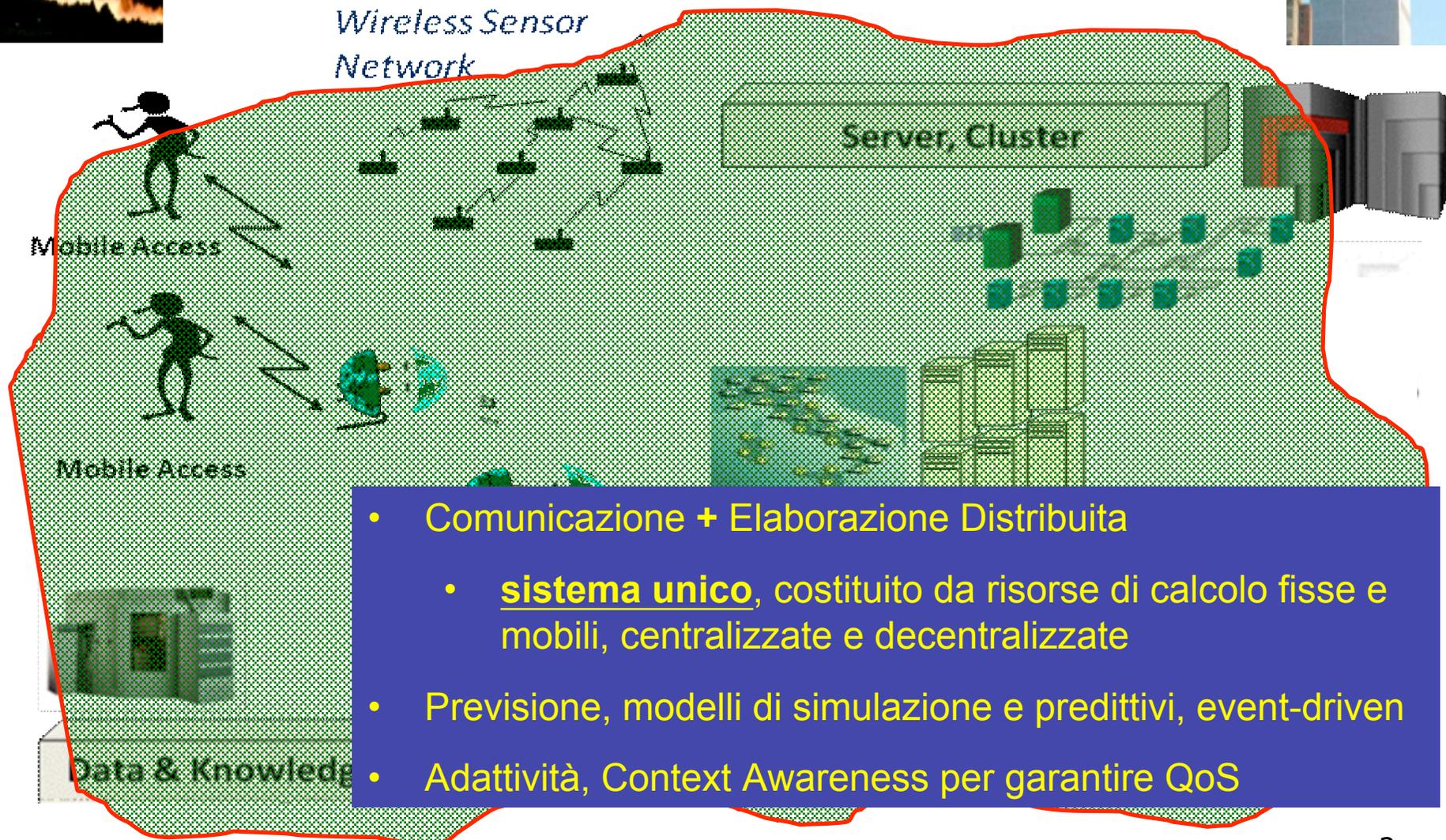
Modello di Programmazione Parallela per
Applicazioni Adattive e Context-Aware

Gabriele Mencagli
Dipartimento di Informatica
Università di Pisa

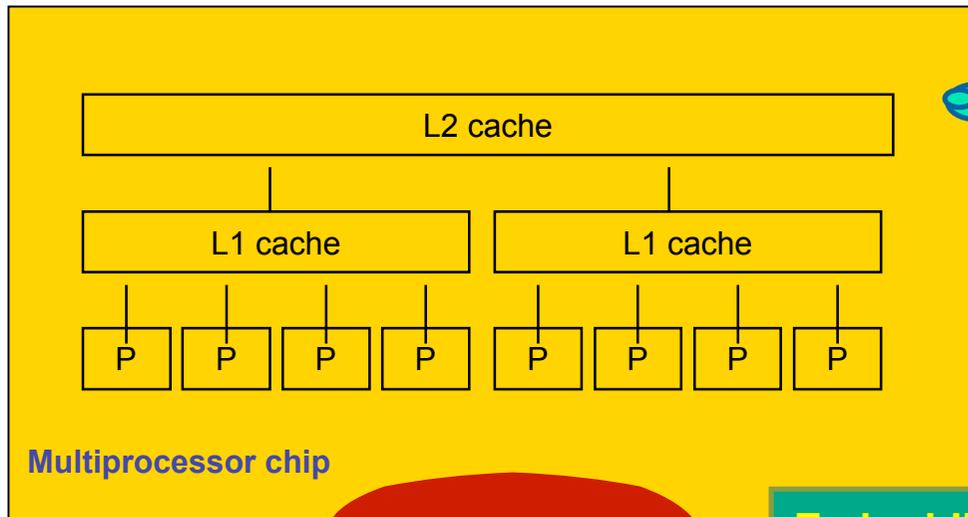
Lo scenario di riferimento



Scenario tecnologico ed applicativo



Evoluzione tecnologica e multicore



Programmabilità !

PORTABILITÀ

Visione uniforme del parallelismo e della distribuzione su scale diverse:
dispositivo - server - grid ...

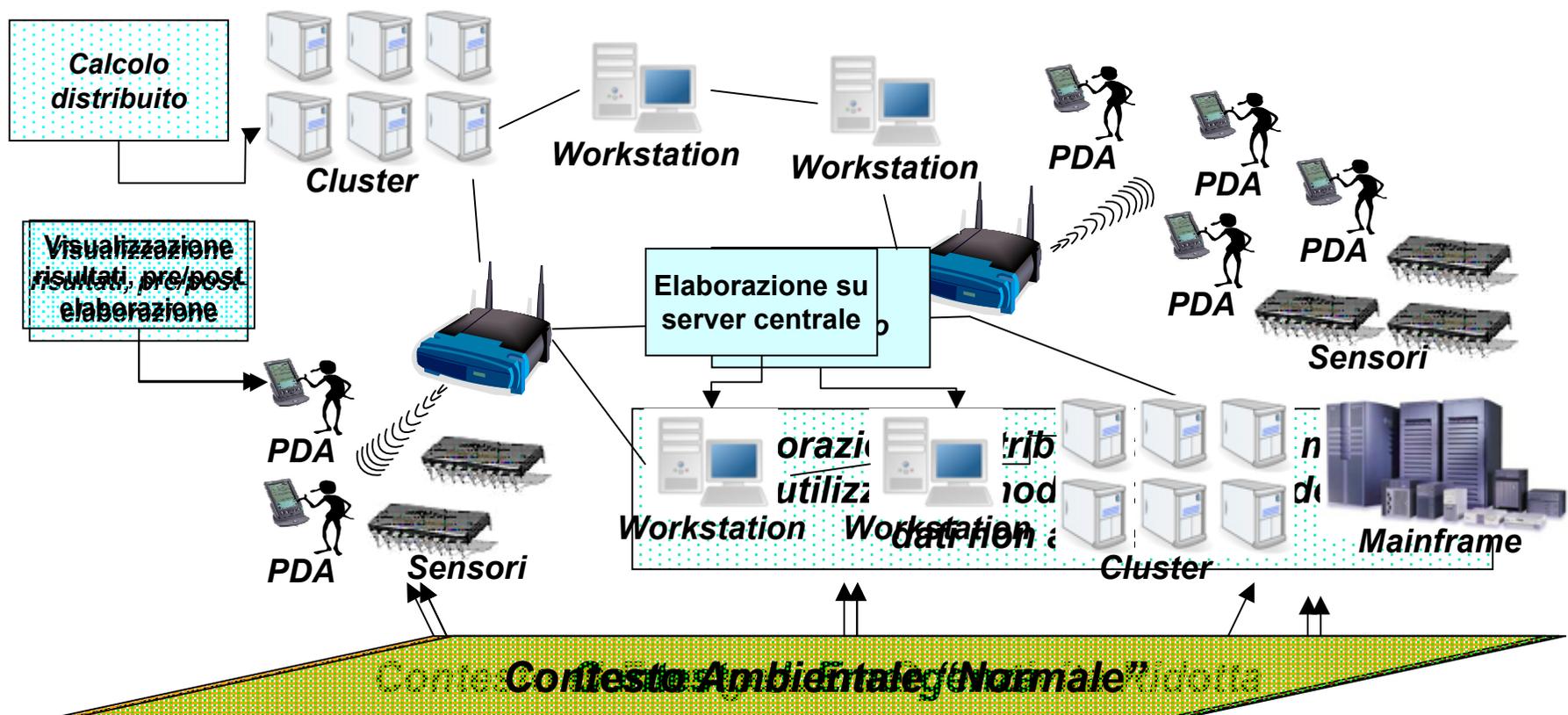
Embedding

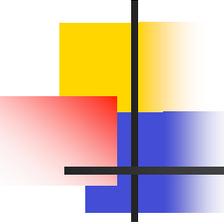
di strutture parallele multicore in dispositivi, sensori, ecc. (in prospettiva ?)

- Elaborazioni decentrate : HPC
- Risorse decentrate come "nodi attivi"
- Network Processors

Le applicazioni adattive e Context-Aware

Situazione di operatività e operatività prioritaria è
Situazione di emergenza, con richieste prestabilite sulla Pervasive
Elaborazione distribuita in rete che ha bisogno della
elevate (QoS), all'applicazione che realizza la previsione.
Personalizza i dati, li segue e li relaziona.





High Performance Pervasive Computing

Gli eventi da dispositivi/sensori e i risultati dei servizi base *possono* innescare **azioni complesse e data/computation intensive**

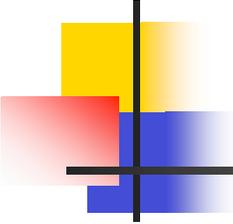
- modelli di previsione e di simulazione
- modelli predittivi dei dati
- supporto alle decisioni (Decision Support System), data mining
- strategie di networking

da rendere **adattive in funzione del "contesto"** (**Context Awareness**).



“nuovo”
settore

***HPPC - High Performance Pervasive
Computing***



Obiettivo: un modello unificante

Applicazioni di In.Sy.Eme per la gestione delle emergenze ambientali (incendi, terremoti, inondazioni, etc...)

come **composizione Adattiva e Context-Aware** di **componenti** di

CALCOLO

e

COMUNICAZIONE

senza schemi rigidi a priori e separazioni nette

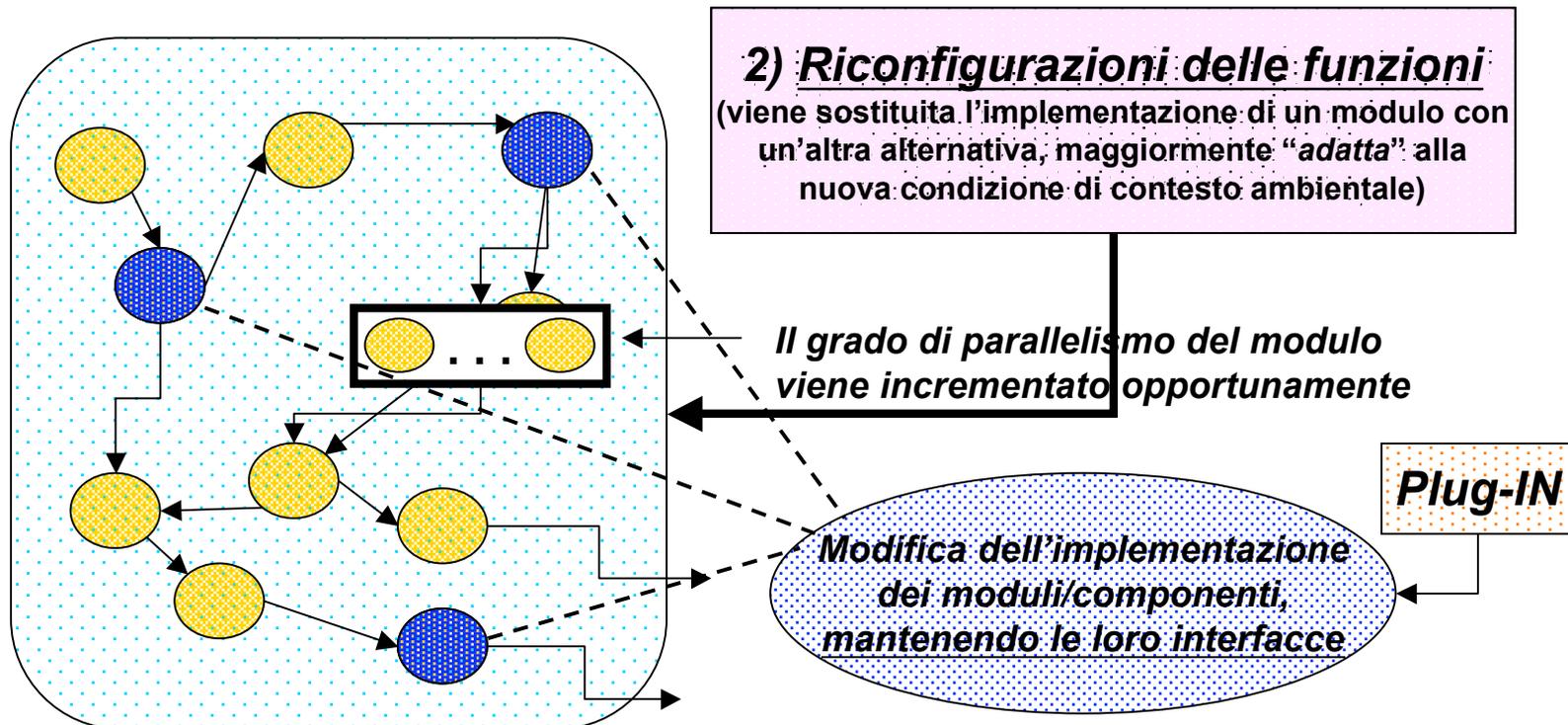
Obiettivo scientifico e tecnologico di grande innovazione

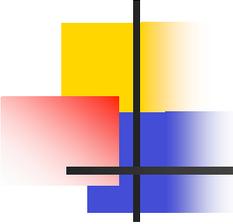
Parallelismo + Adattività + Context Awareness possono entrare in gioco nella strutturazione e progettazione delle strategie

tanto di CALCOLO quanto di COMUNICAZIONE

Le riconfigurazioni di applicazioni adattive

- Lo sviluppo di applicazioni Context-Aware propone due tematiche fondamentali:
 - L'interazione tra i moduli o componenti dell'applicazione e le informazioni sul contesto.
 - L'esecuzione di **operazioni di riconfigurazione** sui moduli o componenti dell'applicazione, in relazione a specifici eventi di contesto.





Context Awareness

Informazioni di contesto:

■ **Eventi**

- *da dispositivi, sensori, etc...*
- *informazioni per la localizzazione spaziale e temporale*

■ **Monitoring**

- comunicazioni (stato delle connessioni), nodi e dispositivi (stato dei nodi centrali e decentrati)
- prestazioni (tempi di servizio, completamento, bande, latenze, etc..)

Context modeling:

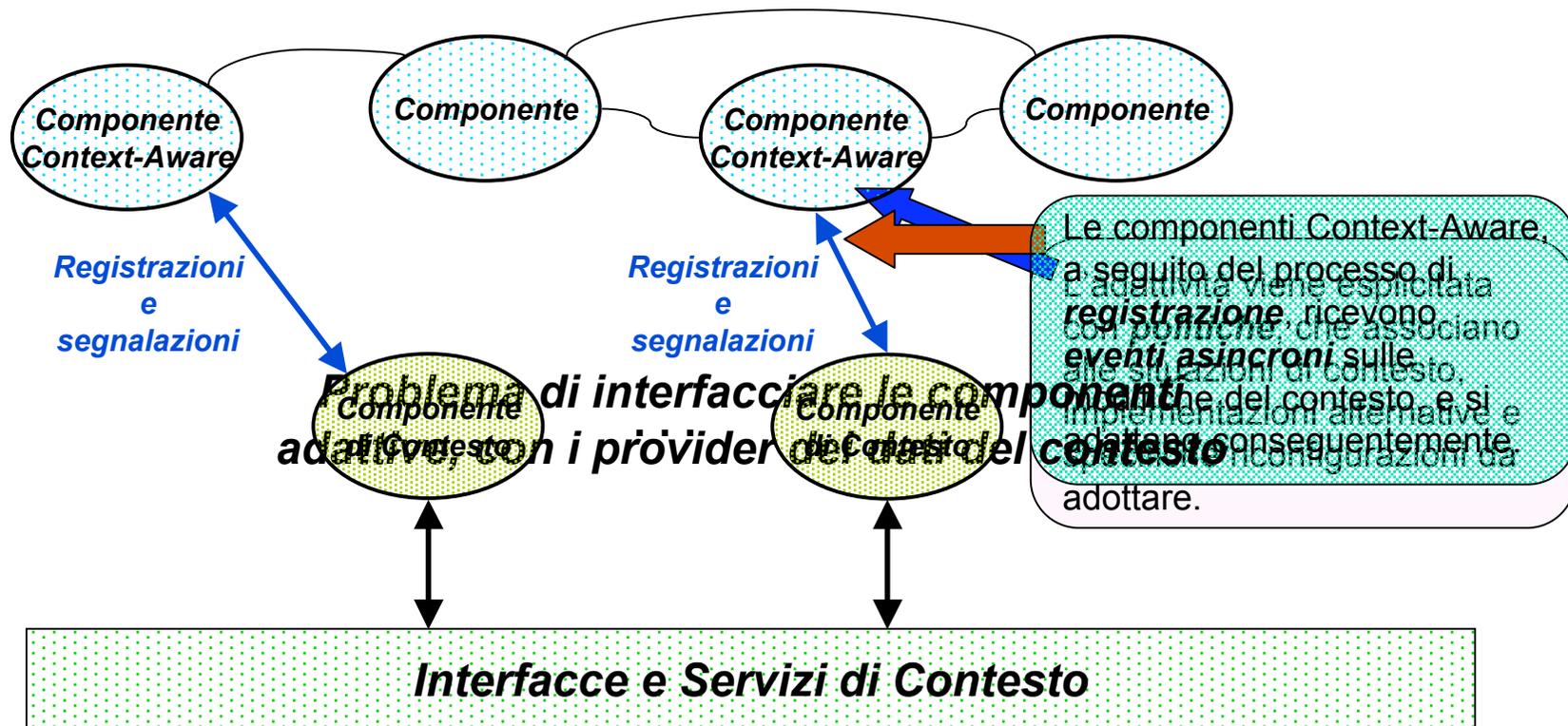
- *Descrizione del contesto ambientale di riferimento, mediante un opportuno modello semantico dei dati ricevuti dalle interfacce di contesto.*

Context reasoning:

- ***inferire*** informazioni non espresse esplicitamente dai dati di contesto disponibili (informazioni e strategie di alto livello su dati dai sensori, controllare e risolvere inconsistenze nei dati e comportamenti, etc...)

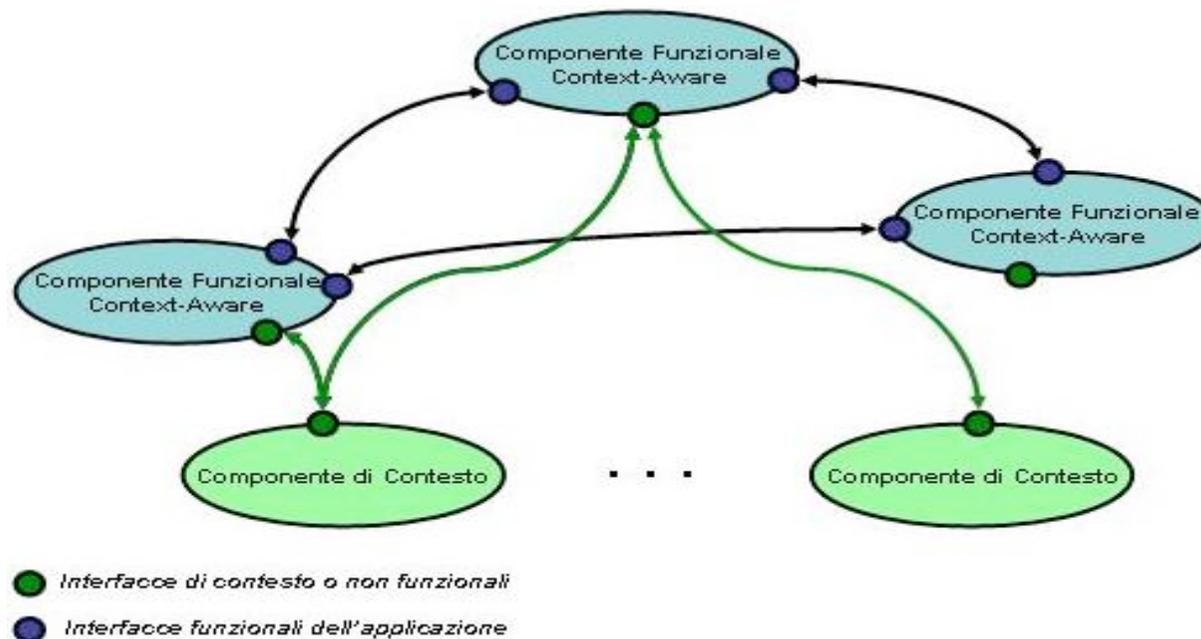
Una prima idea per affrontare la Context Awareness (1)

- **Obiettivo:** un *modello di programmazione parallela* che consenta di esplicitare le dipendenze dal contesto ambientale (*eventi* e *riconfigurazioni*).



Una prima idea per affrontare la Context Awareness (2)

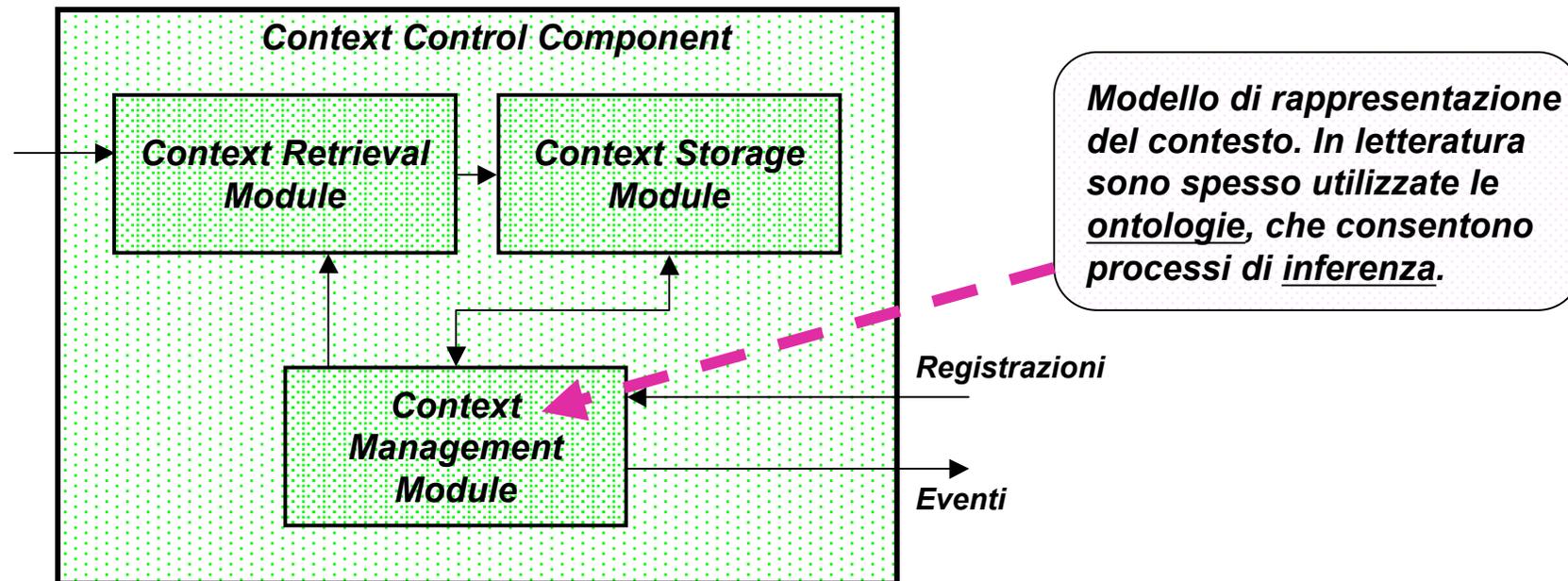
- Le componenti Context-Aware espongono ***interfacce funzionali***, per la loro interconnessione, e ***interfacce di contesto***, per le registrazioni e le segnalazioni con le componenti di contesto.
- Necessità di “***disaccoppiare***” la gestione dei dati di contesto dal ***meccanismo di adattività*** implementato dalle componenti adattive.



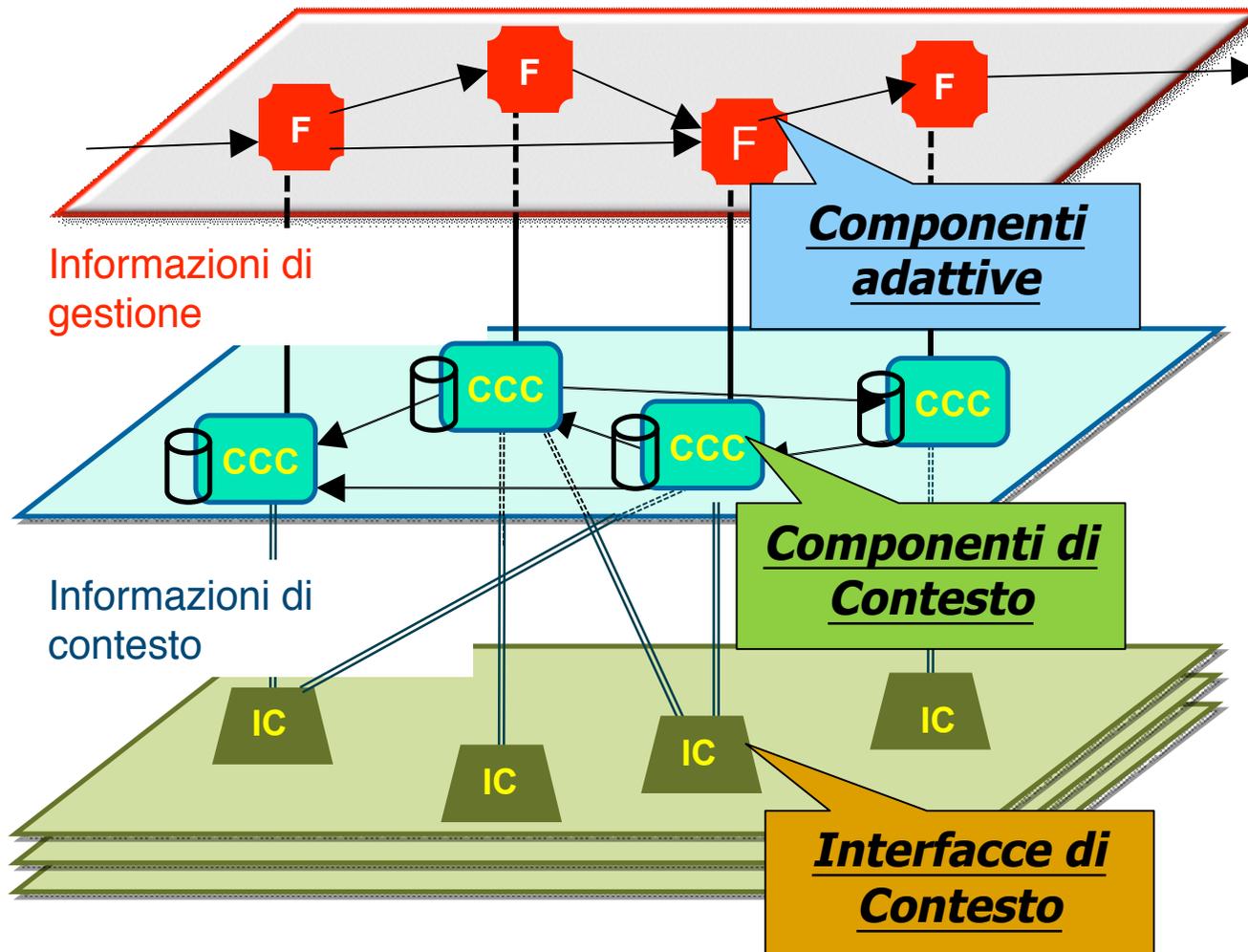
Una prima idea per affrontare la Context Awareness (3)

Attività di:

1. **Acquisizione** dei dati di contesto.
2. **Memorizzazione** dei dati di contesto
3. **Gestione** dei dati di contesto, ed estrazione di ***conoscenza implicita*** mediante attività di ***reasoning***.
4. **Interazione** con altre componenti di contesto e con le componenti funzionali delle applicazioni.



Una prima idea per affrontare la Context Awareness (4)



Grafo di componenti HPC:

livello astratto dell'applicazione; programmazione a componenti + modello HPC.

Rete di Componenti di Contesto:

livello intermedio per la gestione dell'Adattività e Context Awareness.

Contesto fisico:

sensori, reti di comunicazione, nodi, dispositivi, ..., operatori, e relative interfacce di contesto.

Modellazione del contesto ambientale

Descrizione dell'ontologia utilizzata nell'esempio di applicazione per la gestione delle emergenze di inondazione

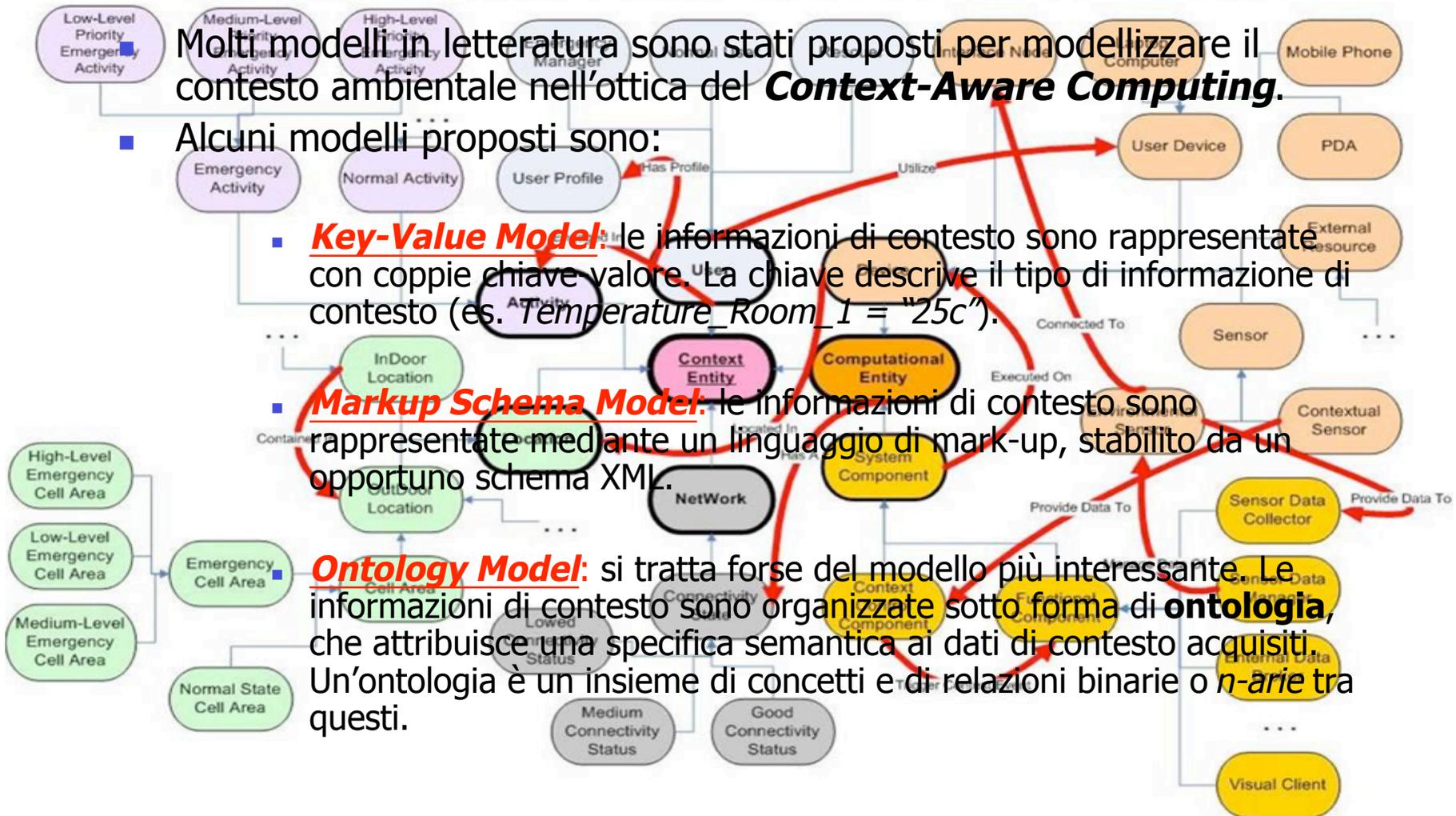
Molti modelli in letteratura sono stati proposti per modellizzare il contesto ambientale nell'ottica del **Context-Aware Computing**.

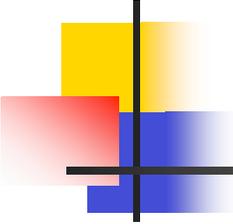
Alcuni modelli proposti sono:

- **Key-Value Model:** le informazioni di contesto sono rappresentate con coppie chiave-valore. La chiave descrive il tipo di informazione di contesto (es. *Temperature_Room_1 = "25c"*).

- **Markup Schema Model:** le informazioni di contesto sono rappresentate mediante un linguaggio di mark-up, stabilito da un opportuno schema XML.

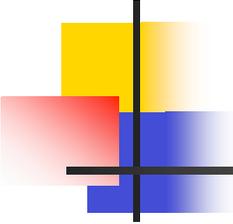
- **Ontology Model:** si tratta forse del modello più interessante. Le informazioni di contesto sono organizzate sotto forma di **ontologia**, che attribuisce una specifica semantica ai dati di contesto acquisiti. Un'ontologia è un insieme di concetti e di relazioni binarie o *n-arie* tra questi.





Modello di programmazione

- **Componenti ad alte prestazioni, portabili, con deployment dinamico**
- **Adattività e Context Awareness**
- **Permeabilità dei livelli e contratti dinamici tra livelli**
- **Nel complesso: un modello per garantire la *QoS* a tutti i livelli in maniera integrata.**



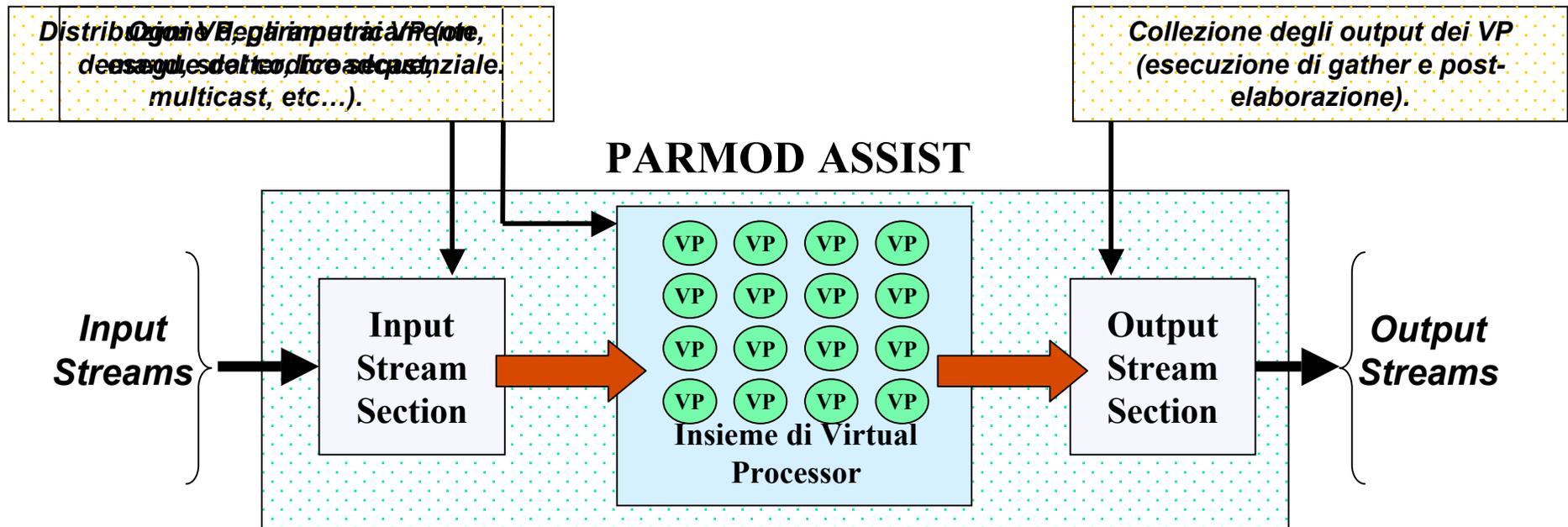
Necessità di utilizzo di ASSIST

Utilizzato con successo in diversi campi di applicazione

- **Computational Chemistry, and other scientific codes,**
 - **Image & Signal Processing,**
 - **Earth Observation Systems,**
 - **Video Compression,**
 - **Knowledge Discovery and Data Mining, User Profiling, Workflow,**
 - **Search Processing on Structured / Unstructured Data,**
 - **Query Language Interpreters, ...**
-
- Prestazioni paragonabili o superiori a quelle ottenute con librerie di basso livello (MPI), ma con tempi di sviluppo inferiori di uno o più ordini di grandezza.

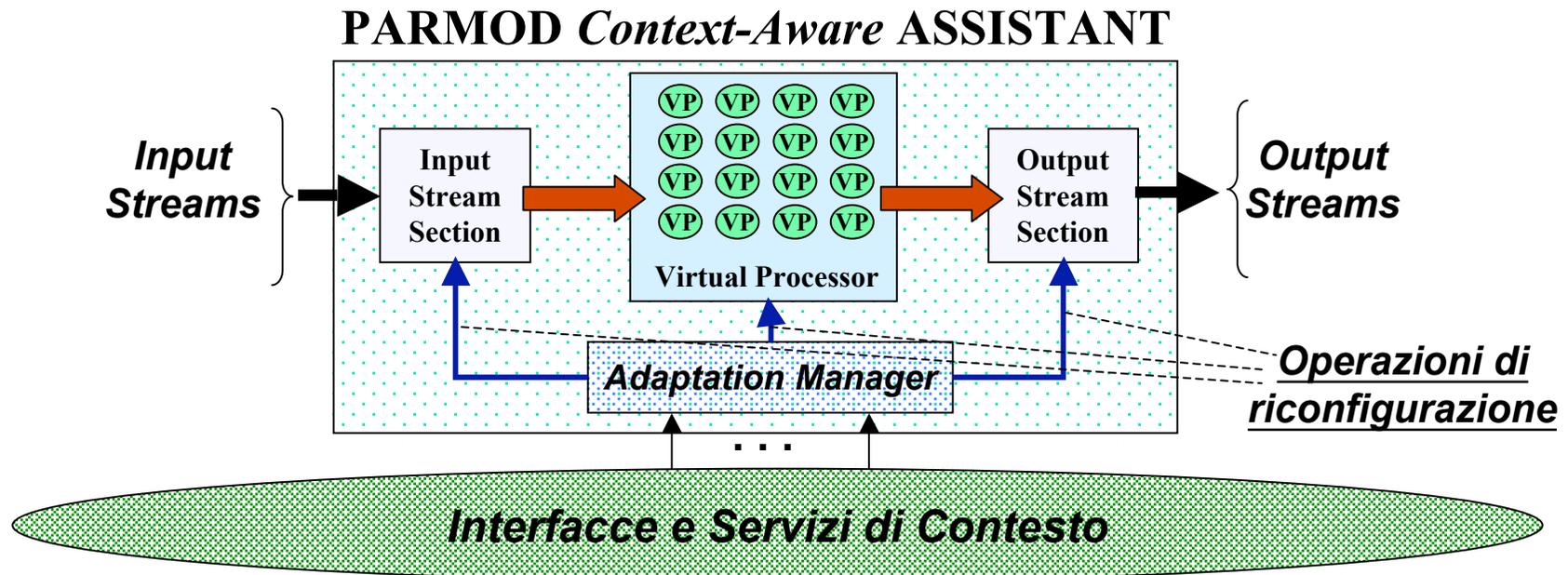
Richiamo al concetto di PARMOD

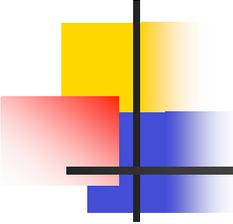
- Necessità di un modello di programmazione ad **alte prestazioni** (parallela) e **Context-Aware** (che esponga meccanismi di adattività al contesto).
- ASSIST introduce un costrutto generico di programmazione parallela, che può di volta in volta istanziare uno specifico *skeleton* parallelo (farm, map, stencil statico, stencil dinamico, etc...). Il costrutto prende nome di **PARMOD** (Parallel Module).



Il PARMOD adattivo di ASSISTANT

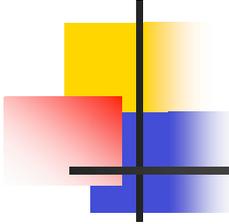
- E' stato introdotto **ASSISTANT** (*ASSIST with Adaptivity and coNText awareness*), un'estensione di ASSIST con l'introduzione del costrutto **PARMOD** adattivo al contesto.
- ASSISTANT prevede l'introduzione dell'**Adaptation Manager**, come nuovo elemento primitivo del PARMOD, insieme all'Input Section, all'Output Section e ai Virtual Processor.
- Il manager è responsabile dell'esecuzione di operazioni di riconfigurazione sul modulo parallelo, in reazione alle modifiche del contesto ambientale a cui è interessato.





Il manager dell'adattività (1)

- Mette in relazione le informazioni di contesto con il flusso dei dati (stream) e le funzionalità del modulo, per realizzare strategie di adattività e Context Awareness:
 - Sceglie le *alternative funzionali*
 - Sceglie le *forme di parallelismo* per tali funzionalità
 - Associa funzionalità, forme e strategie a *tipi di nodi* e di *reti*
- Interazione e cooperazione tra Manager (rete di Manager).
- *Context Awareness, contratti e permeabilità dei livelli*: espressi esplicitamente nella rete degli Adaptation Manager.



Il manager dell'adattività (2)

- ASSIST (versione 1.3) implementa un primo meccanismo di dinamicità, in grado di modificare esclusivamente il numero di **processi VPM** su cui sono “mappati” i VP dei PARMOD.

Riconfigurazioni sulla Input Section: può essere necessario modificare a *run-time* la modalità di distribuzione degli input (on demand, scatter, broadcast) e di pre-elaborazione.

Input
Stream

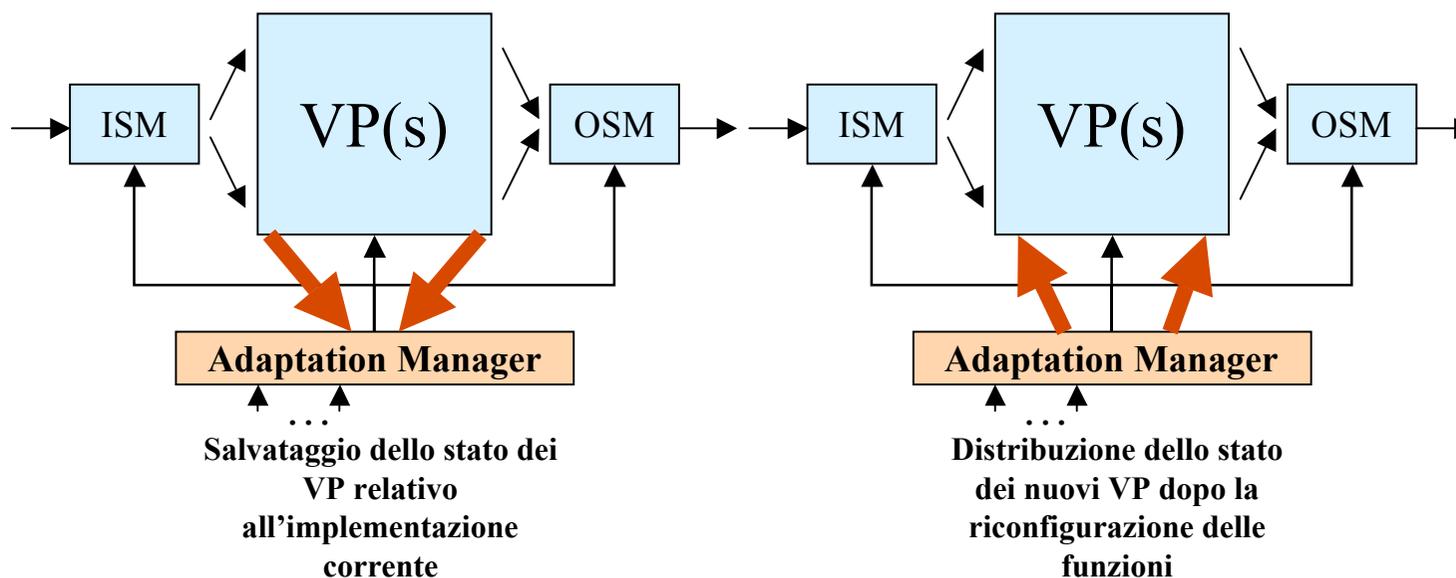
Riconfigurazioni sulla Output Section: può essere necessario modificare il collezionamento dei dati prodotti dalle elaborazioni dei Virtual Processor. Le modalità di *gather* e di post-elaborazione possono essere riconfigurate.

Riconfigurazioni sui VP: può essere possibile modificare il codice sequenziale eseguito dai VP, utilizzando un'alternativa funzionale “adatta” alla nuova situazione ambientale e alle nuove risorse disponibili.

In ASSISTANT questo tipo di riconfigurazione è solo un caso particolare!

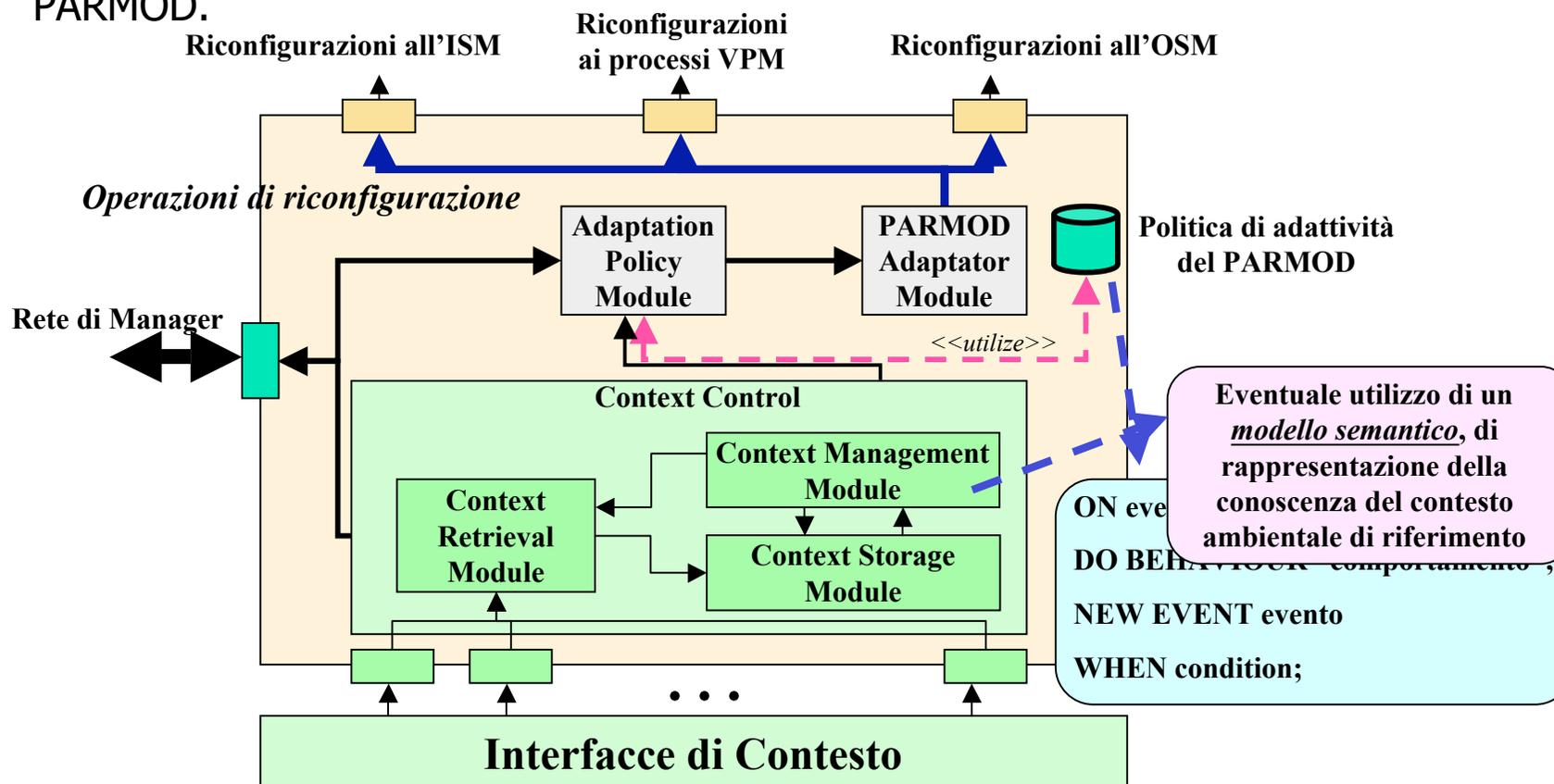
Il manager dell'adattività (3)

- Nel caso di **riconfigurazioni delle funzioni**, il manager sceglie le alternative funzionali del modulo, può modificare il codice sequenziale eseguito dai VP, modificare la forma di parallelismo, etc...
- Il manager può avere la necessità di gestire lo stato associato alle diverse implementazioni alternative e ripristinarlo in modo consistente quando necessario.



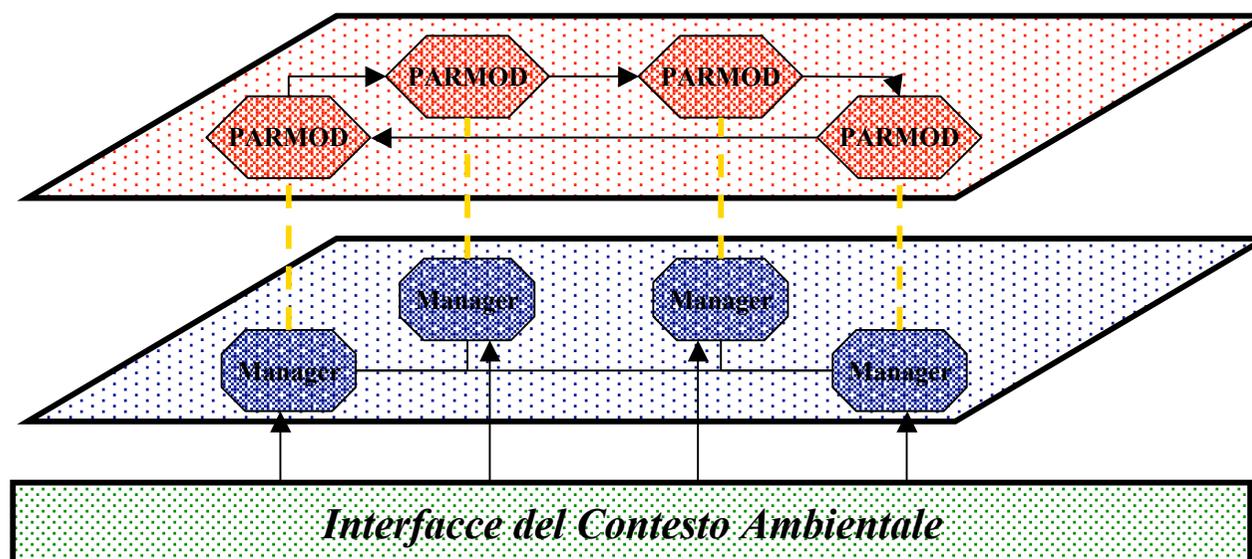
Il manager dell'adattività (4)

- L'Adaptation Manager** è responsabile del processo di acquisizione, memorizzazione e gestione dei dati del contesto, e delle riconfigurazioni del PARMOD.

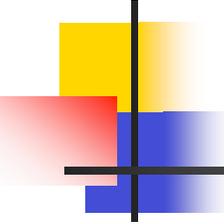


Il manager dell'adattività (5)

- Un'applicazione parallela è strutturata da più moduli sequenziali e paralleli interconnessi, ciascuno con il suo Adaptation Manager connesso in rete con gli altri (*rete di manager*).



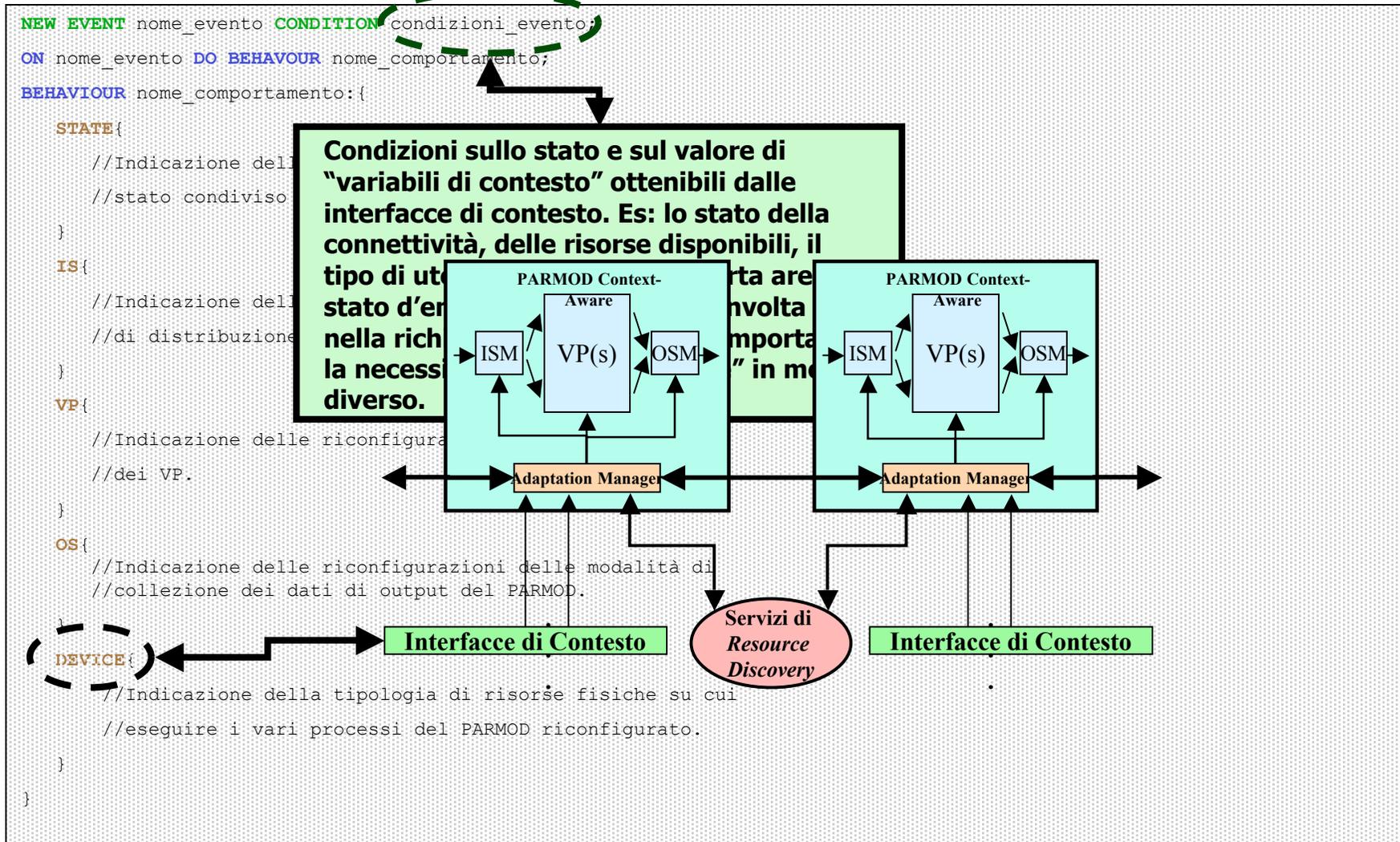
- La rete di manager è necessaria per un duplice scopo:
 1. Realizzare processi di **riconfigurazione globale** di più moduli dell'applicazione.
 2. Realizzare la **condivisione dei dati del contesto** ambientale tra diversi moduli dell'applicazione (**Contesto Partizionato**).



La politica di adattività del manager (1)

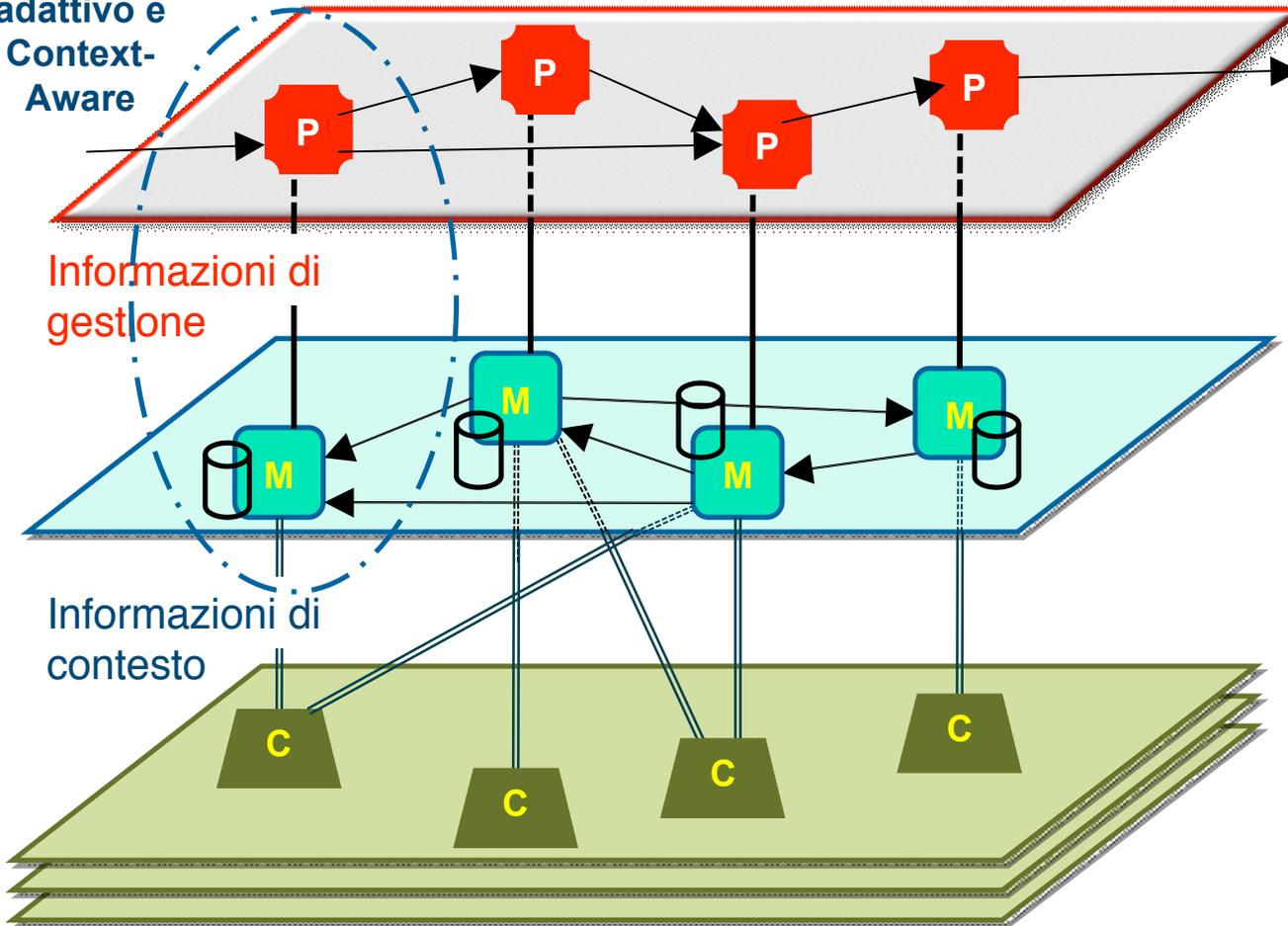
- **Le implementazioni alternative di un PARMOD devono essere espresse con una sintassi specifica, che le colleghi ad eventi di contesto. Con questa sintassi deve essere possibile, quindi, programmare l'Adaptation Manager di un PARMOD adattivo.**
- **Ciò avviene con le regole di politica di adattività dell'Adaptation Manager.**
- **Il supporto di ASSIST deve realizzare un unico modulo parallelo, in grado di "attivare" le diverse implementazioni, quando si verificano gli eventi corrispondenti.**
- **Le alternative possono essere fornite dinamicamente (meccanismo di plug-in).**

La politica di adattività del manager (2)



Applicazione a livelli ("Piani")

Componente
adattivo e
Context-
Aware



Grafo di moduli paralleli:

livello astratto dell'applicazione; programmazione a componenti + ASSIST (o altro modello HPC).

Rete di Manager:

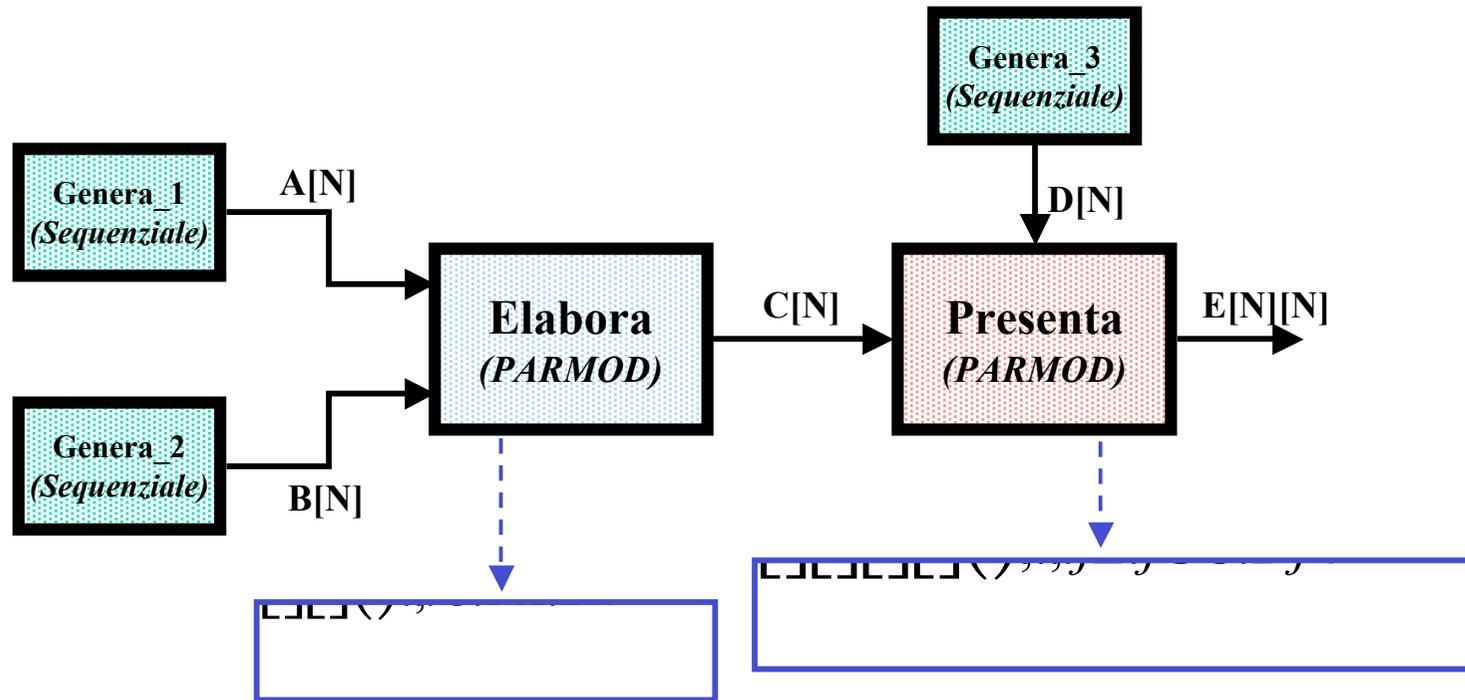
livello intermedio per la gestione dell'adattività e Context Awareness.

Contesto fisico:

sensori, attuatori, reti di comunicazione, nodi, dispositivi, ..., operatori, e relative interfacce di contesto.

Un esempio di applicazione adattiva (1)

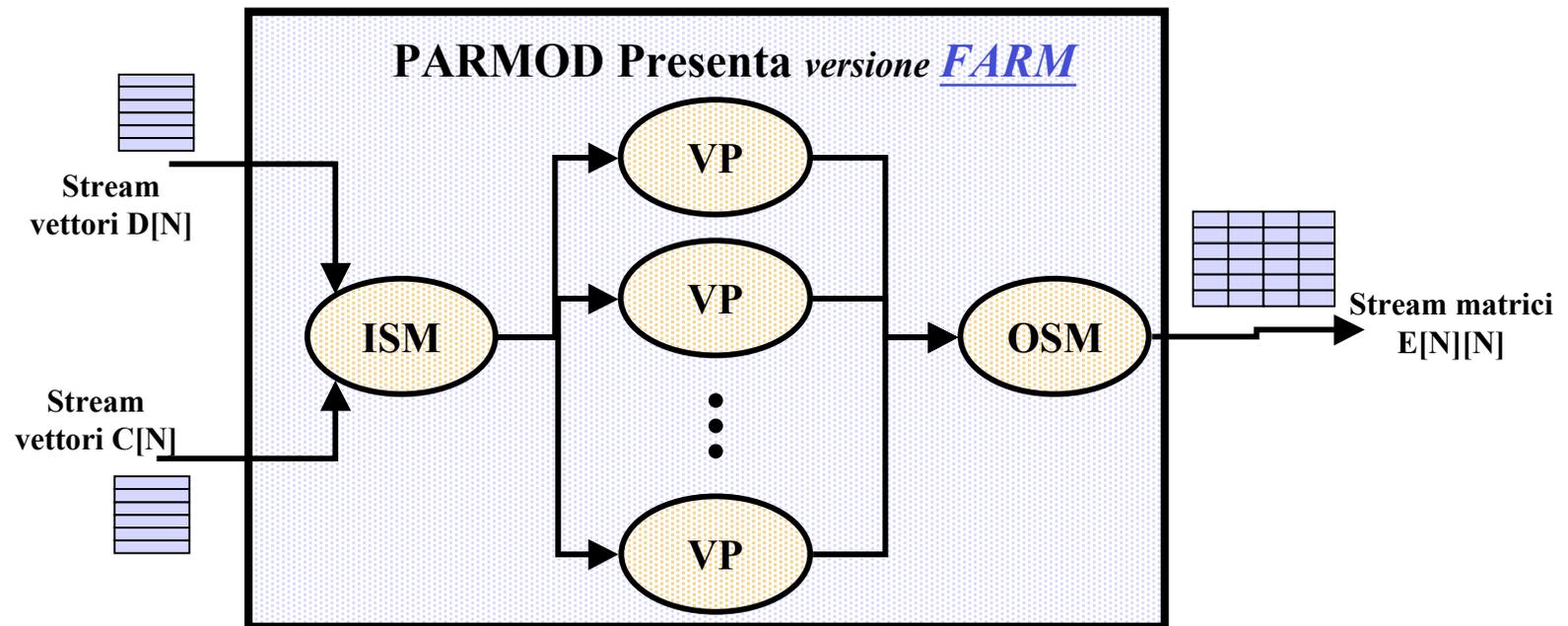
- Esempio di applicazione parallela adattiva, realizzato emulando i meccanismi che saranno realizzati con ASSISTANT nei prossimi mesi.



- L'applicazione di esempio non richiede il mantenimento di uno stato nelle diverse attivazioni dei PARMOD (*caso esemplificativo*).

Un esempio di applicazione adattiva (2)

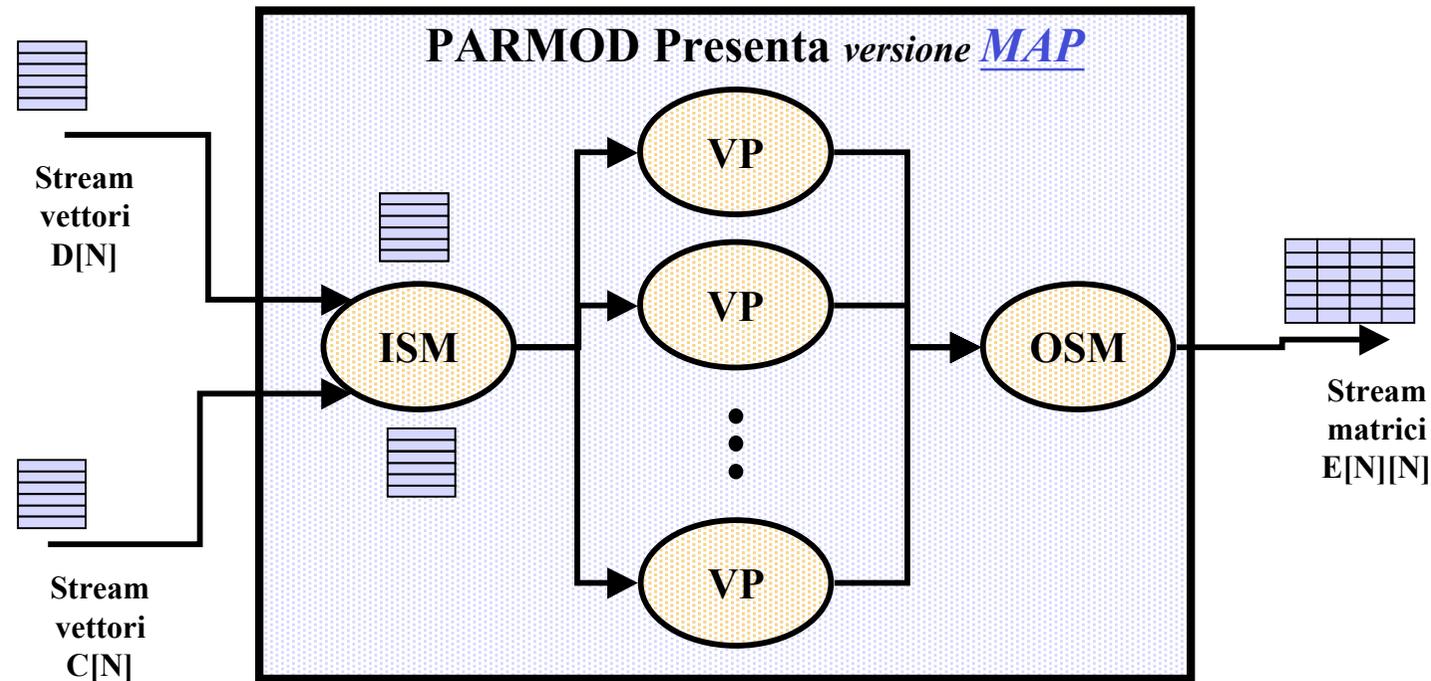
- I moduli paralleli Elabora e Presenta possono avere implementazioni parallele alternative, nel nostro caso una **task-parallel FARM**, l'altra **data-parallel MAP**.



Ogni VP calcola per intero la matrice E. I vettori di input vengono distribuiti on demand ai VP, da parte del processo ISM.

Un esempio di applicazione adattiva (3)

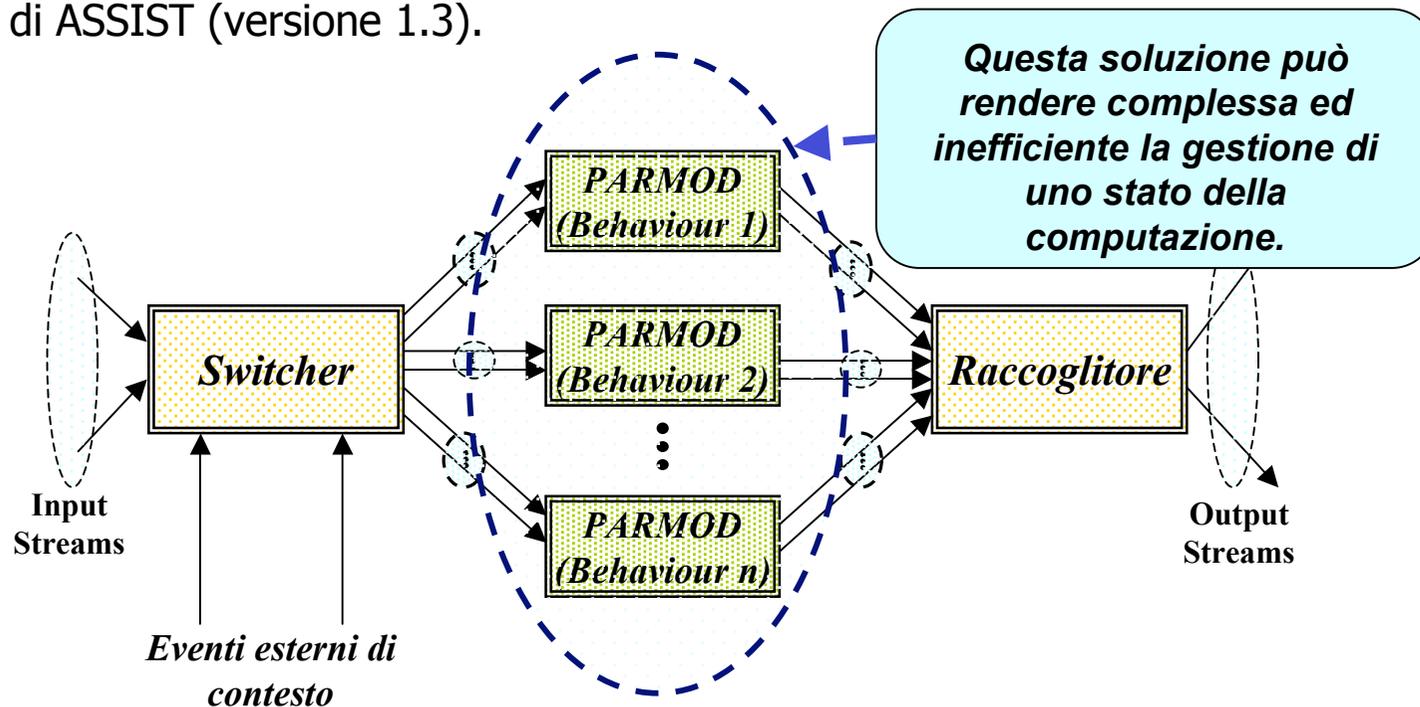
- La **MAP** comporta un partizionamento delle strutture dati e può essere preferibile in quelle situazioni in cui l'utilizzo di memoria sia un aspetto critico.



Ogni VP calcola indipendentemente una riga della matrice E. Viene eseguita la broadcast di D e la scatter di C.

Emulazione di un PARMOD adattivo (1)

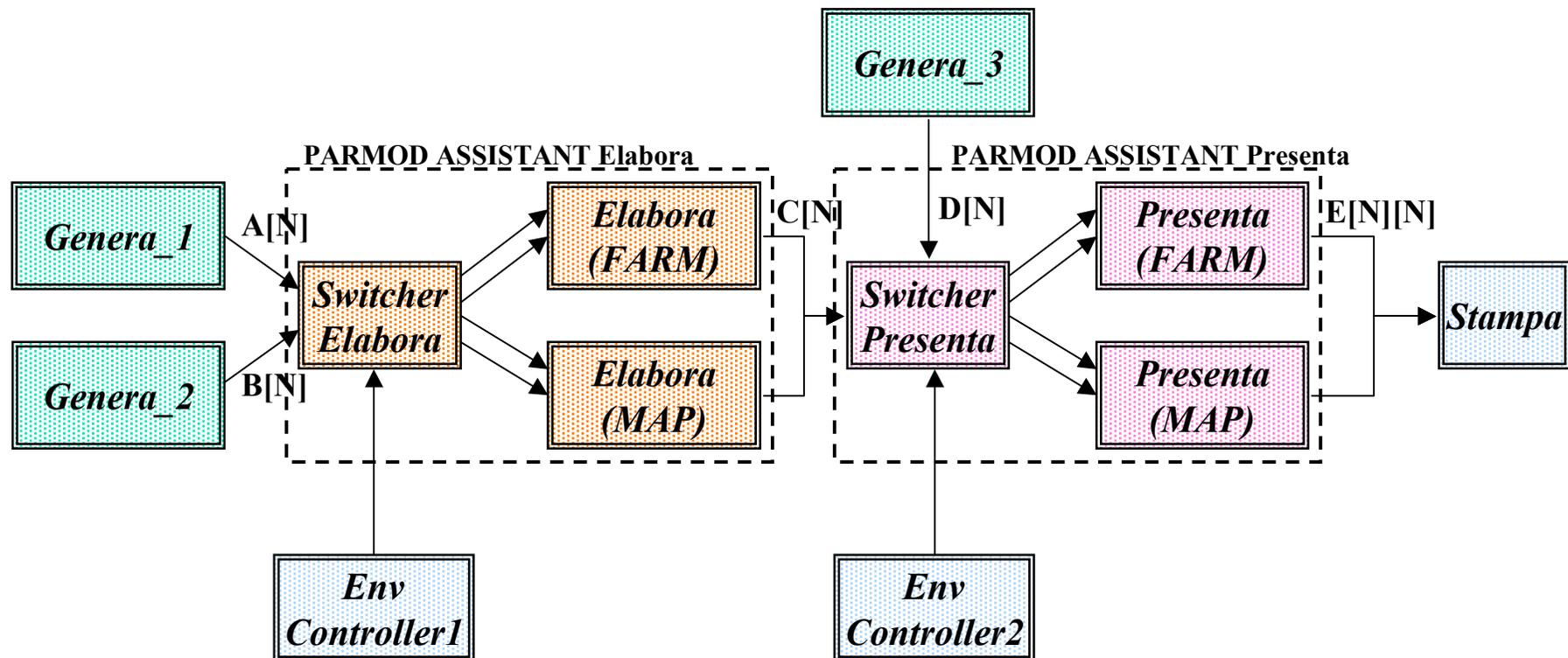
- Le riconfigurazioni sul PARMOD ASSISTANT sono state **emulate** con gli strumenti attuali di ASSIST (versione 1.3).



- L'emulazione più intuitiva consiste **nel definire un PARMOD specifico per ogni alternativa funzionale**. Questo è probabilmente l'unico modo di procedere, se si utilizza un linguaggio parallelo non strutturato come **MPI**.

Emulazione di un PARMOD adattivo (2)

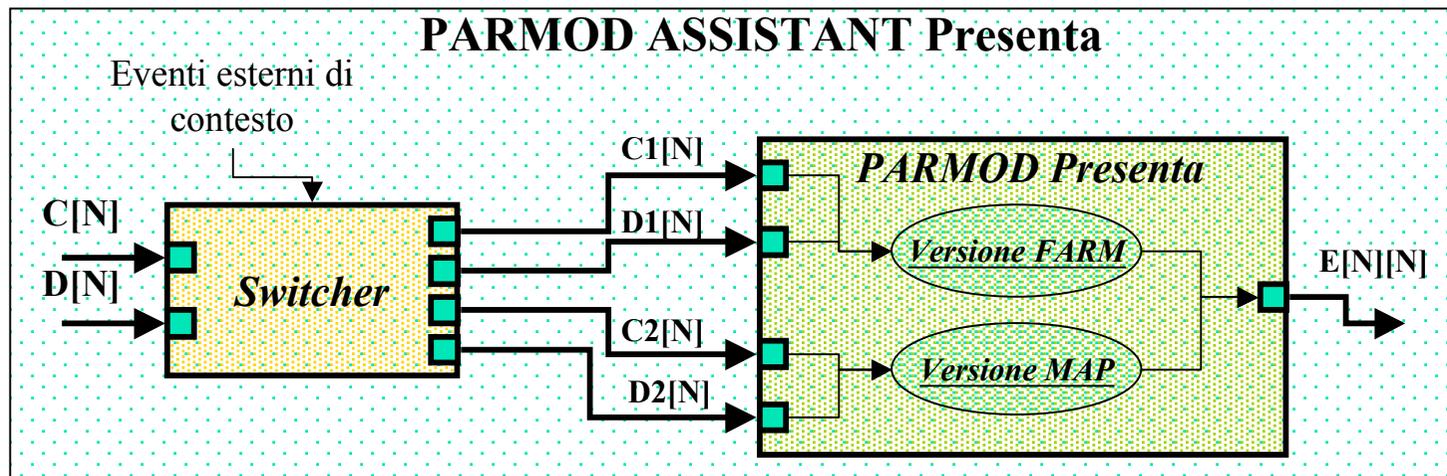
- Di seguito illustriamo lo schema della prima emulazione dell'applicazione adattiva.



- Gli eventi di contesto sono simulati dai moduli *EnvController1* e *EnvController2*.

Emulazione di un PARMOD adattivo (3)

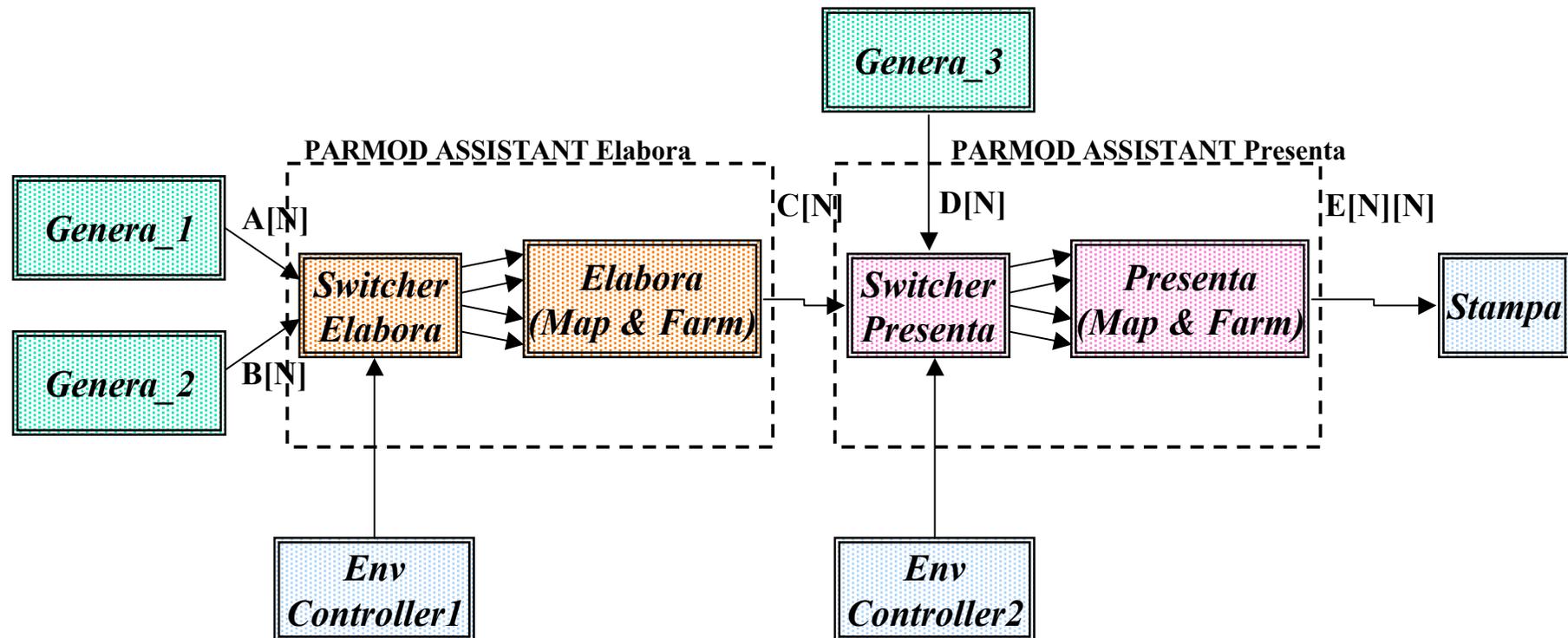
- Nell'emulazione proposta, si sfrutta al massimo la proprietà del costrutto PARMOD di descrivere una generica computazione parallela.
- Un **PARMOD adattivo e Context-Aware** si realizza mediante un unico modulo parallelo, i cui VP eseguono alternative funzionali staticamente definite.



- Le diverse alternative funzionali vengono attivate in base agli stream su cui il PARMOD riceve.

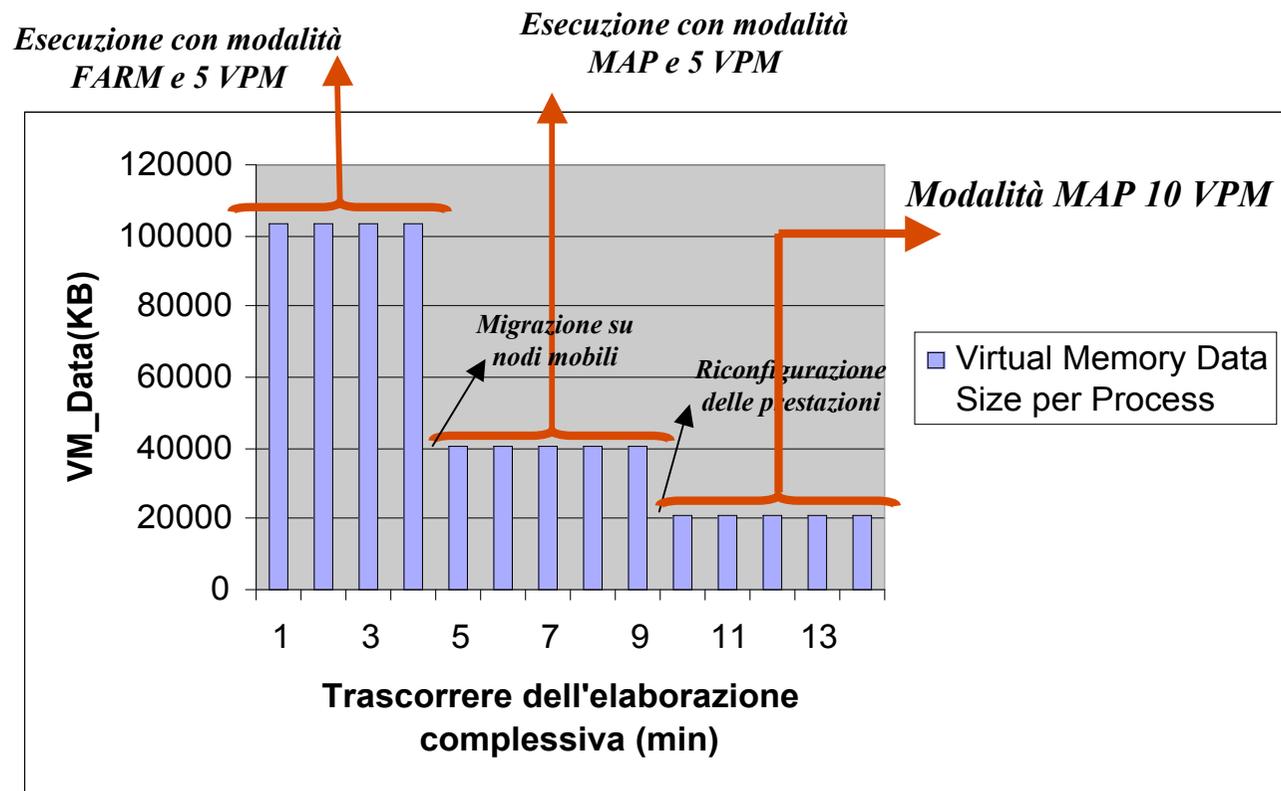
Emulazione di un PARMOD adattivo (4)

- Di seguito illustriamo lo schema della seconda emulazione dell'applicazione adattiva.



Alcuni risultati delle emulazioni (1)

- Riportiamo di seguito alcuni risultati dei test sul **PARMOD ASSISTANT Presenta**, eseguiti sul cluster "**Pianosa**" presso il Dipartimento di Informatica.



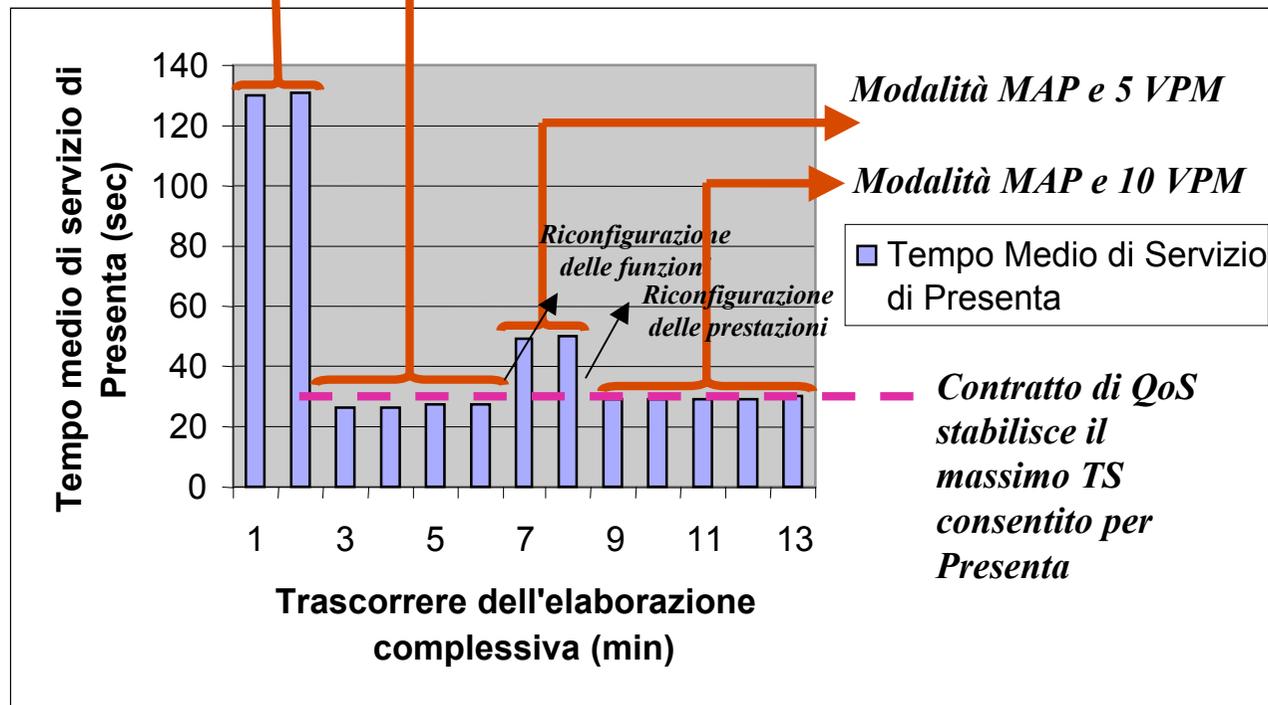
- Dimensione di N = **5000** interi
- Occupazione N_ = 25M interi ~ **95 MB**
- Il calcolo della matrice E richiede un tempo medio di elaborazione di **120 secondi**.

Alcuni risultati delle emulazioni (2)

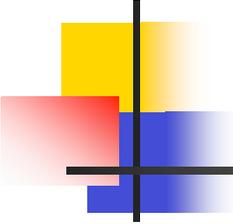
- Riportiamo di seguito alcuni risultati dei test sul **PARMOD ASSISTANT Presenta**, eseguiti sul cluster "**Pianosa**" presso il Dipartimento di Informatica.

Caso iniziale con modalità FARM e 1 VPM

Modalità FARM e 5 VPM



- Dimensione di N = **5000** interi
- Occupazione N_ = 25M interi ~ **95 MB**
- Il calcolo della matrice E richiede un tempo medio di elaborazione di **120 secondi**.



Conclusioni e sviluppi futuri

- L'estrema eterogeneità delle risorse e la dinamicità dell'ambiente, comporta la necessità di un modello di programmazione parallelo e Context-Aware.
- **Le applicazioni di INSYEME devono sapersi adattare autonomamente al contesto, modificando le loro implementazioni o riconfigurandosi per rispettare contratti di performance.**
- Il lavoro nel futuro riguarderà la completa realizzazione dei meccanismi ASSIST di nuova generazione (ASSISTANT), con un linguaggio che consenta di:
 1. **Definire i moduli dell'applicazione e le loro alternative funzionali.**
 2. **Collegare le alternative funzionali ad eventi di contesto ambientale (*politiche di adattività dei manager*).**
 3. **Eseguire le due operazioni del punto precedente dinamicamente, oltre che prevederle staticamente.**



Domande ?

