data { 1 DE — "centralized" {name, value} | multiple DE — "distributed" }

(variables in PL; records/tables in DB; virtual address in OS regions) ↔ (memories: regs, caches, RAM, disks)

(i)

Control of execution { 1 PE — "centralized" {id, exec. state} | multiple PE — "distributed" {locus of control} }

(Threads) → (processor elements)

Mappings

Dynamic data

- : values change in t of a DE
- : DE created/destroyed
- : Mappings Log → Phy change in t

Dynamic control

- : exec. state always changes in t
- : PE created/destroyed
- : Mappings Log → Phy change in t

1 Log. DE → Multiple Phys. DE (Replicas / Partitions)

Multiple Log. DE → 1 Phys. DE [Sharing Phys Resources]

1 Log. PE → Multiple Phys. PE (Replication of Control)

Multiple Log. PE → 1 Phys. PE [Time Multiplexing]

<transparently>
<isolation between Logical contexts>

For Distributed data or control units

↳ Composition of DE/PE elements — at Logical or Physical Levels of Abstraction

- : can be uniform → multiple DE aggregated as a collection
  → multiple PE aggregated as a ~~work~~pool of workers or a cluster of processors
- : or mixed → combinations of DE + PE <the basic "Execution Units"?>
  → concept of OS Process
  → concept of PL Object
  → concept of Component
  → concept of Service
  → concept of Agent
  → concept of a VM

Logical combinations of DE/PE are related

- to concept of Locality → spatial and temporal distance or proximity concepts > what is a Location or a Place? Logical / Physical
- How do Mappings Log → Phy preserve / affect the Locality characteristics of a Logical organization

↳ **Interactions / Communication** [again Consider this at Logical and Physical Levels]

- Structure/Topology: between elementary ~~DE~~ and PE units or compound units
- Behavior/Semantics of Interaction: access to a unit: through entry points < name/address, port, method — single access point {how are these handled by the unit?}, multiple access points
- Can be Static / Dynamic
- Communication mechanisms: direct {messages, RPC/RMI} | collective {broadcast, multicast group} | indirect {shared memory (Logical or Physical), software bus (MOM, Queueing Systems, Events Pub/Sub)}

p4. | concept of Centralized/Decentralized — ? a broader concept ?

1.1 / really interesting set of case study applics. — ? felt that 2.15 Applic. brief summaries / fine which are
<albeit too long a section> <2.14>
would help right at the end of each appl. description
perhaps

<narrative description combined w. a more ? syntactic ? taxonomy-oriented ?> (as proposed in 4. Vectors Extra) ✓ (DPA paper of)

3 – | dynamism + distribution | — ? very nice effort to quantify the properties

p35 3.1. is OK – it would be nice to try a kind of taxonomy !

? wonder if this could benefit from a clearer separation of concerns:
- External data
- application characteristics / Logic / control
- Mappings (cf Figure 2, p41) ? seems to mix the Logical and the Physical)

seem are critically important sections as they relate to kind of taxonomy concepts

3.2 Types of distribution

3.3.1 / metrics – very interesting / important
p44

Issues:

a) . at application level

Application characteristics

- (logical) DE and PE
- their (compositions)
- their (logical interactions) <structure / behavior>
- (how all the above change in time) ← Dynamics

- the functional operators acting upon them (2)
- the non-functional spec

b) . at "system level" ?

- how are [Mappings] performed ?

= or multiple layers here ||

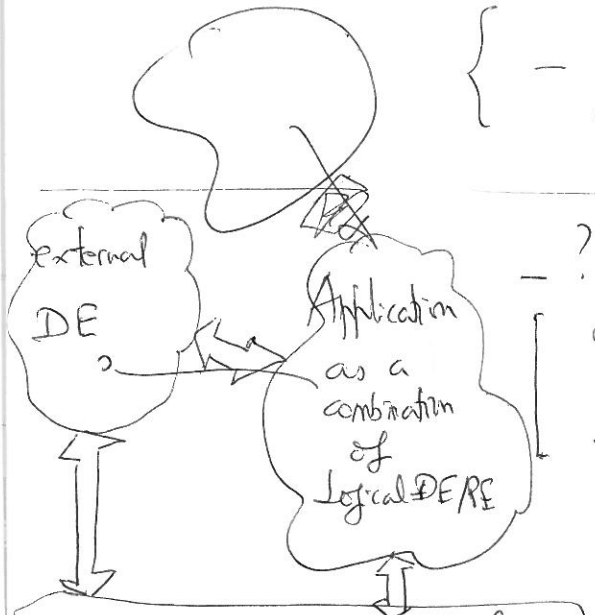
c) . at physical infrastructure level

- (Physical)
- likewise a)

Dynamics

A data-centric perspective

- focus on how DE are organised Logically & Physically <according to application/infrastructure>
- analyse the [Mappings] — evaluate them.
- analyse combinations of DE/PE to improve the Mappings and preserve Locality



- ? think how External Data Elements exist and are generated versus how Data Elements are modeled from within an Application ?