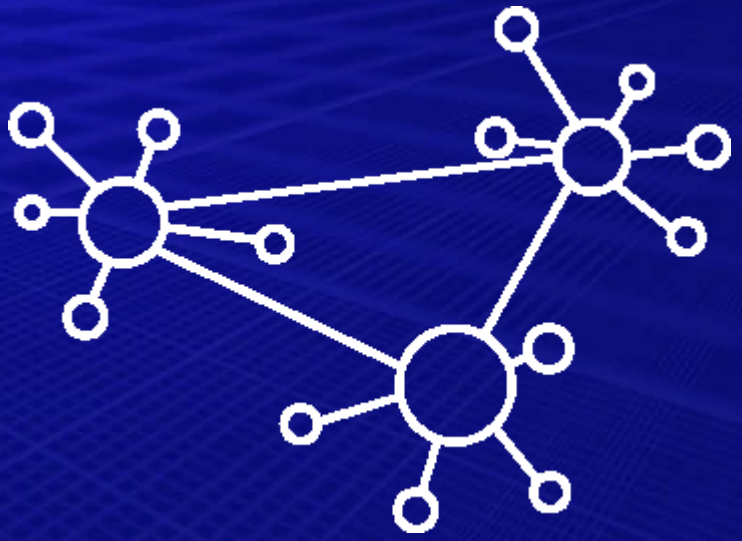
Distributed Associative Networks

Kevin Hartig, Digital Reasoning Systems

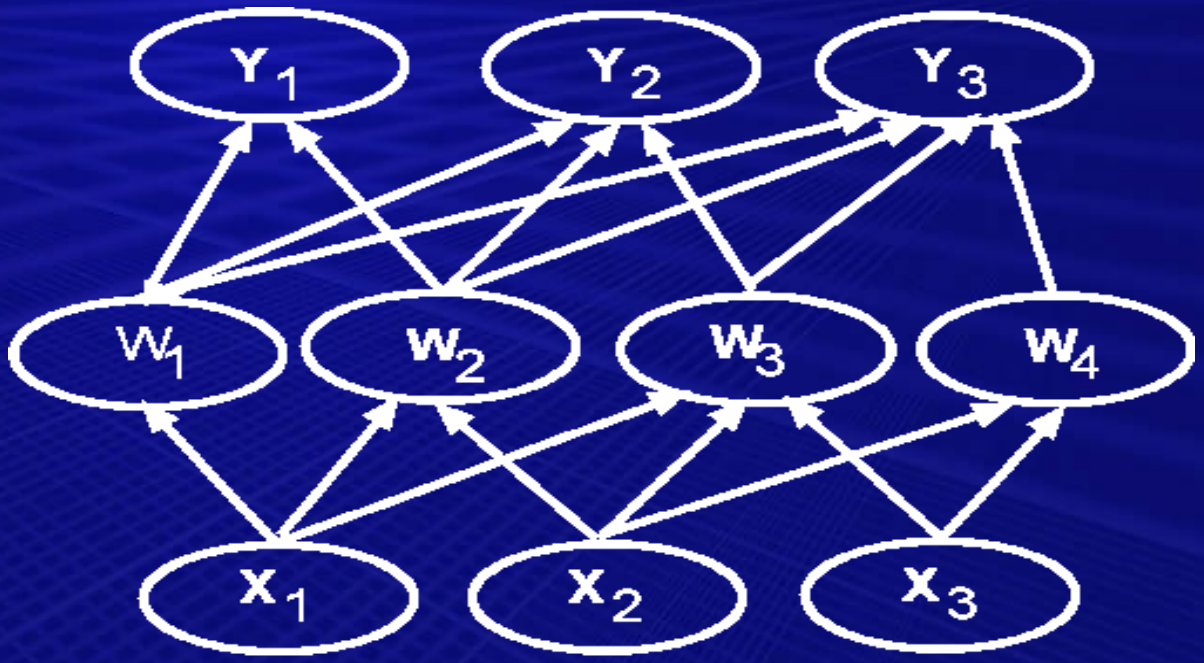


Introduction

- Kevin Hartig
  - Digital Reasoning Systems
  - Director of Distributed Computing
  - Unstructured Data Analytics (UDA) solutions
  - AI and Distributed Computing
  - Finding meaning in the information age

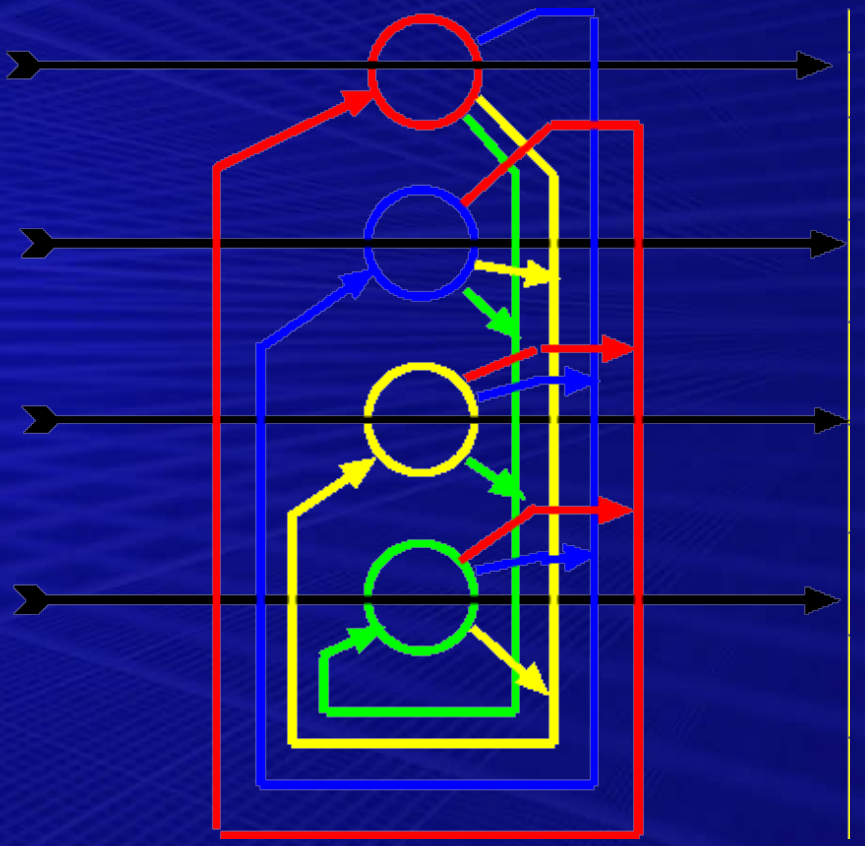
Associative Networks

- Represent abstract models of the brain
- Consists of numerous, highly interconnected computational units
- Pattern -> Classifiers, Associators
- Associative
  - Hetero (HA)
  - Auto (AA)



Hopfield Network

- John Hopfield – 1982
- Auto – Associative
- Outputs feedback into all Inputs but its own
- Weights calculated in advance



Hopfield Network Processing

$$a_i \leftarrow \begin{cases} 1 & \text{if } \sum_j w_{ij} s_j > \theta_i \\ 0 & \text{otherwise} \end{cases}$$

$$E = -1/2 \sum_{i>j} w_{ij} s_i s_j + \sum_i \theta_i s_i$$

$$[W] = \eta [X]^t [Y] \qquad [Y] = [X][W]$$

Hopfield Network Processing

$$[X] = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \qquad [Y] = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

$$[W] = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 2 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 2 & 1 \end{bmatrix}$$

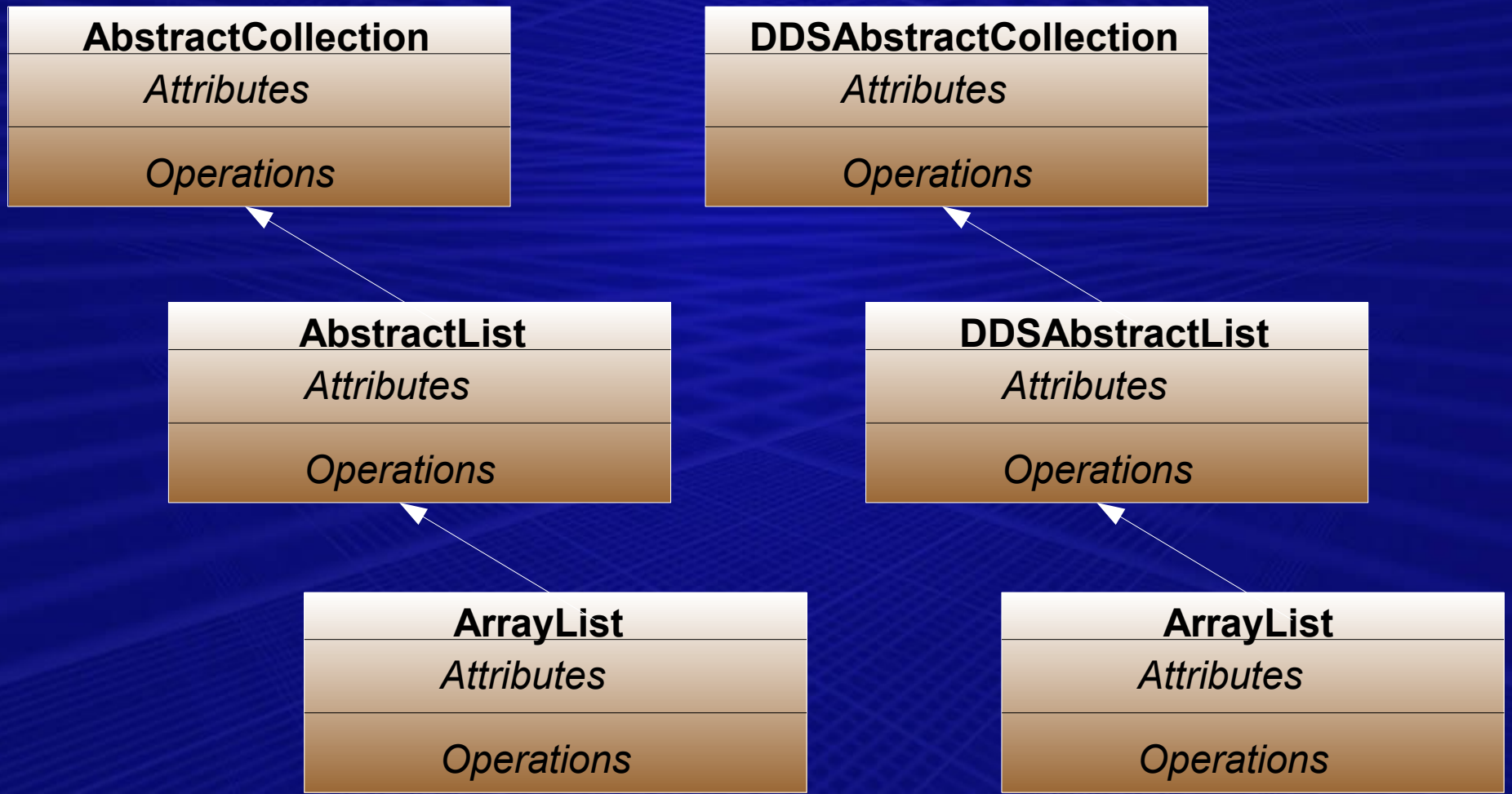
[Y][W] = [Z] apply threshold -> [X]

Distributed Data Structures

- Structure that can be accessed simultaneously by multiple processes
- Represented by a collection of objects
- Independently accessed or modified
- Concurrency rules applied

Distributed Data Structures

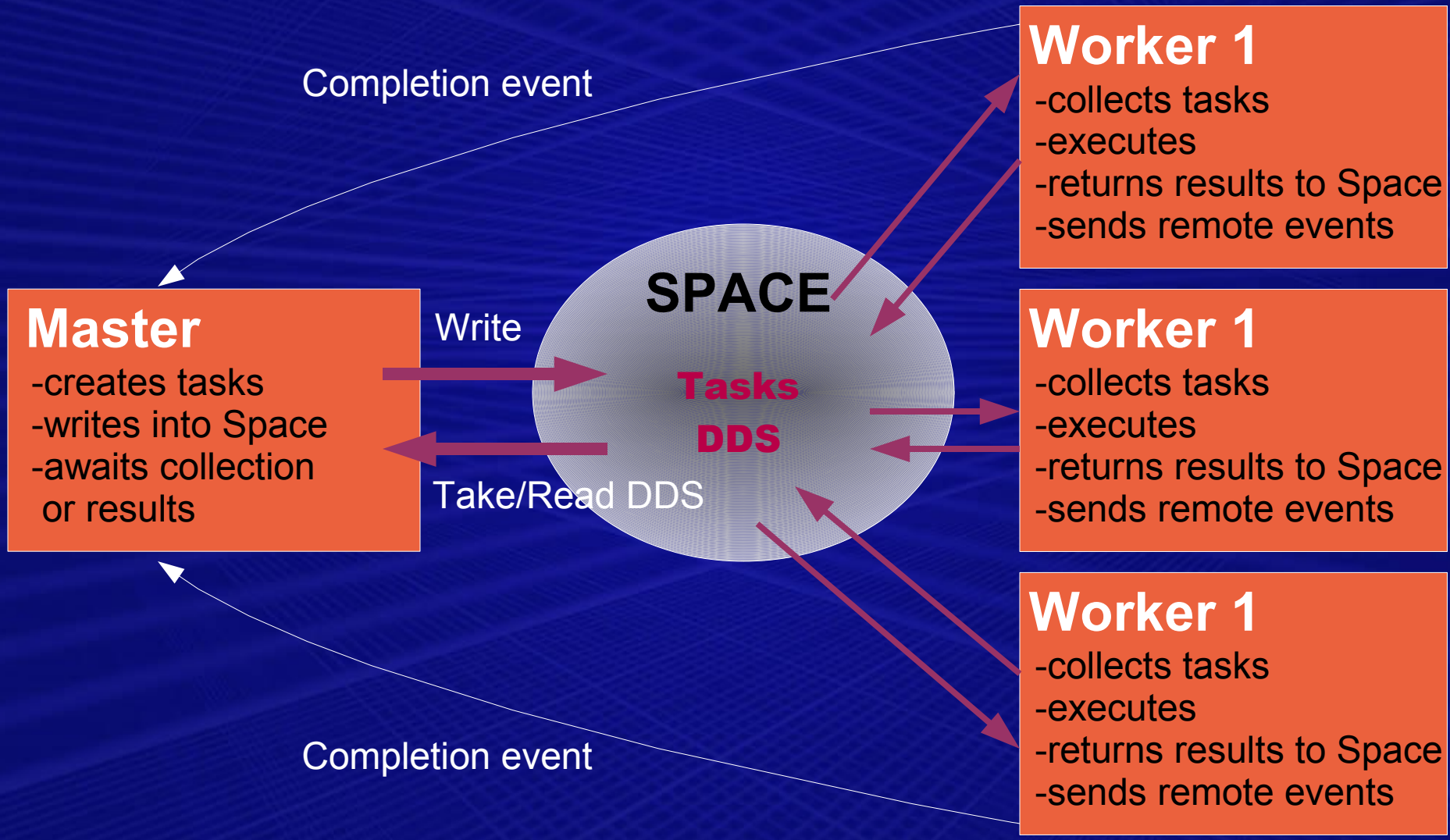
- Follows structure of java.util



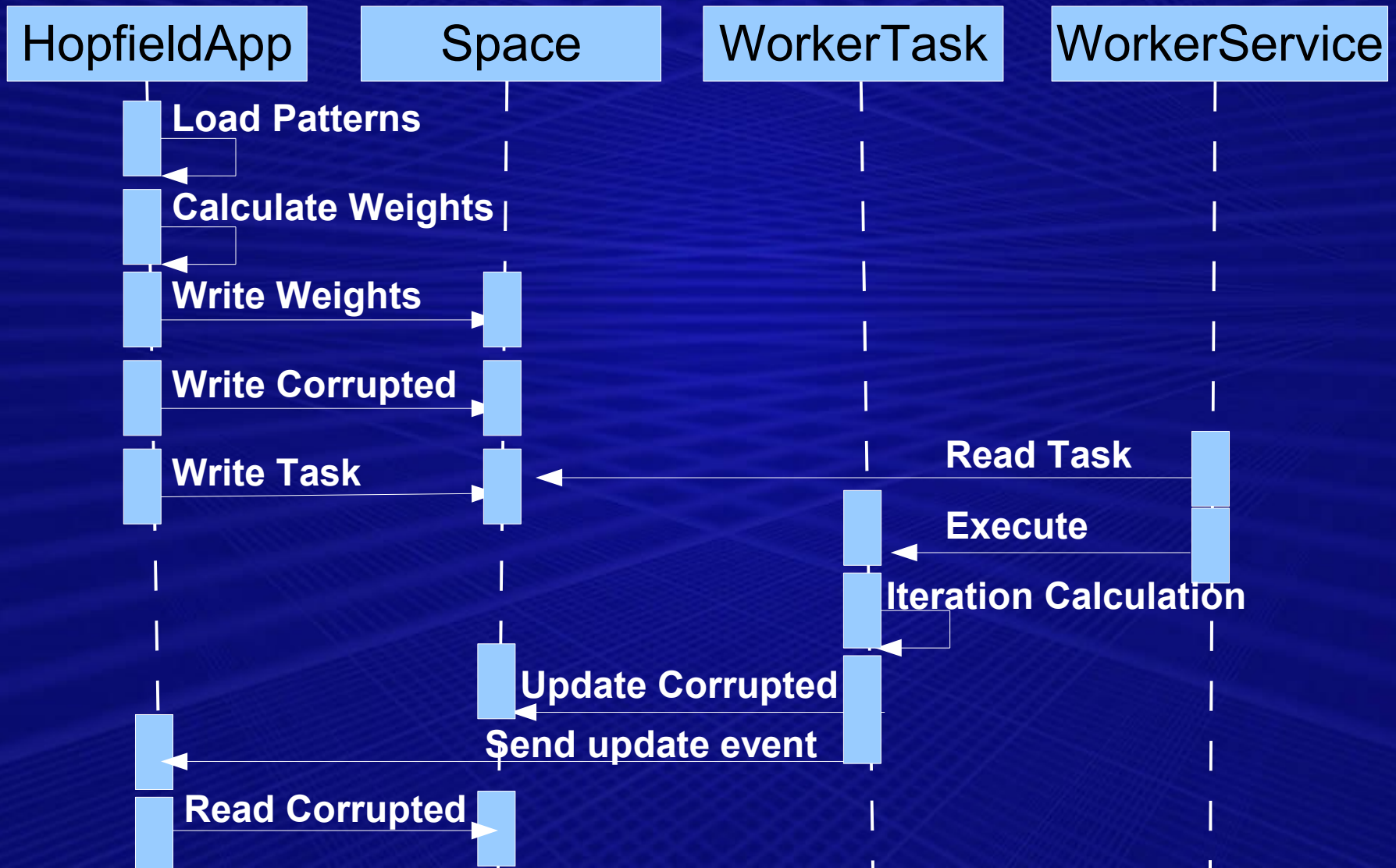
Distributed Data Structures

```
String name = "test array";
DDSArrayList array = new DDSArrayList(name);

for (int i = 0; i < 10; i++) {
    array.add("test " + i);
}

System.out.println("Size of array is " + array.size());

for (int i = 0; i < array.size(); i++) {
    elem = (String) array.get(i);
    System.out.println("Elem " + i + " is " + elem);
}

array.set(5, "Set Test");
array.add(7, "Added element at index 7");
System.out.println(" ");
for (int i = 0; i < array.size(); i++) {
    elem = (String) array.get(i);
    System.out.println("Elem " + i + " is " + elem);
}
```

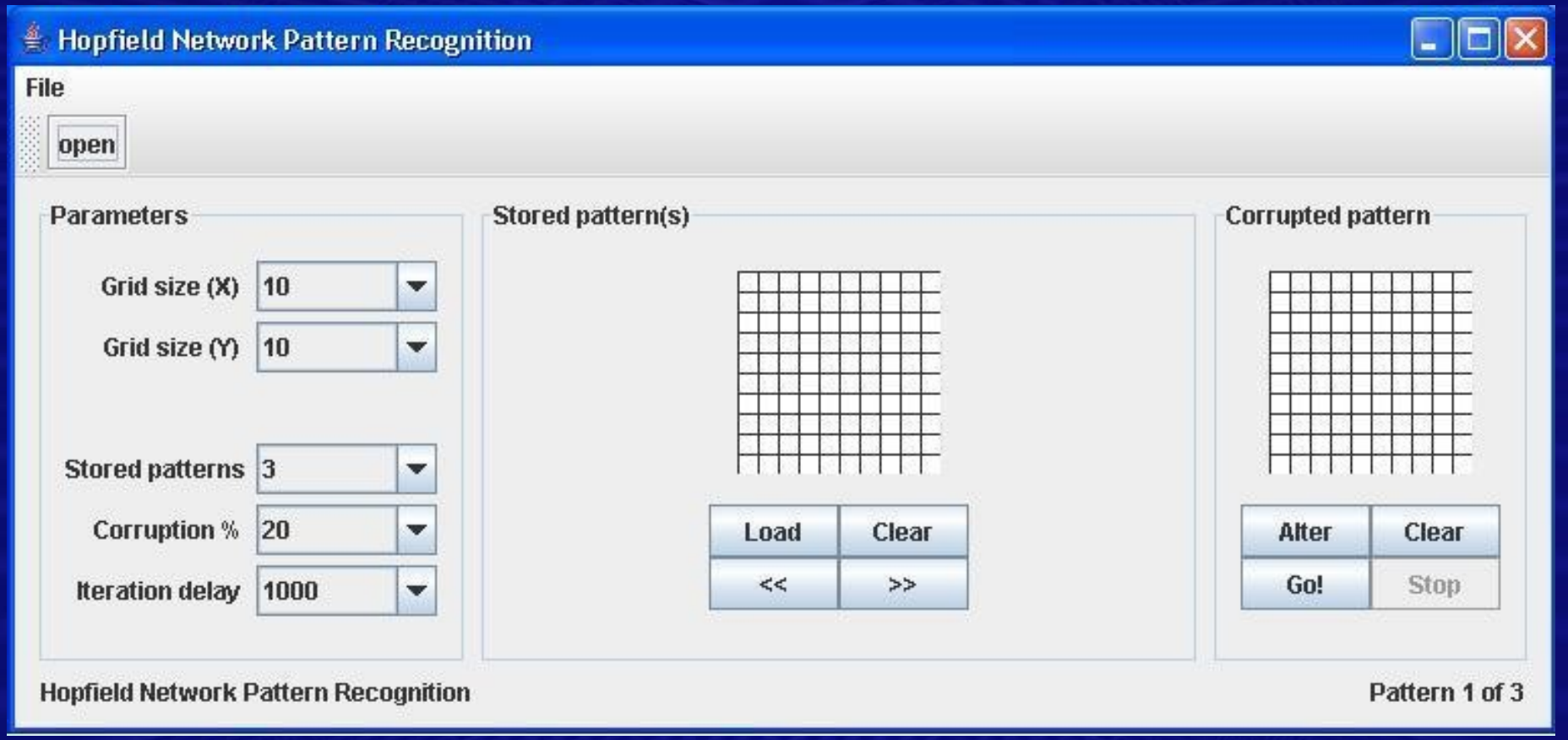
Workflow Architecture



Distributed Design



Hopfield Network Pattern Recognition Example



Contact

- Kevin Hartig
- kevin.hartig@digitalreasoning.com
- kevin.hartig@gmail.com