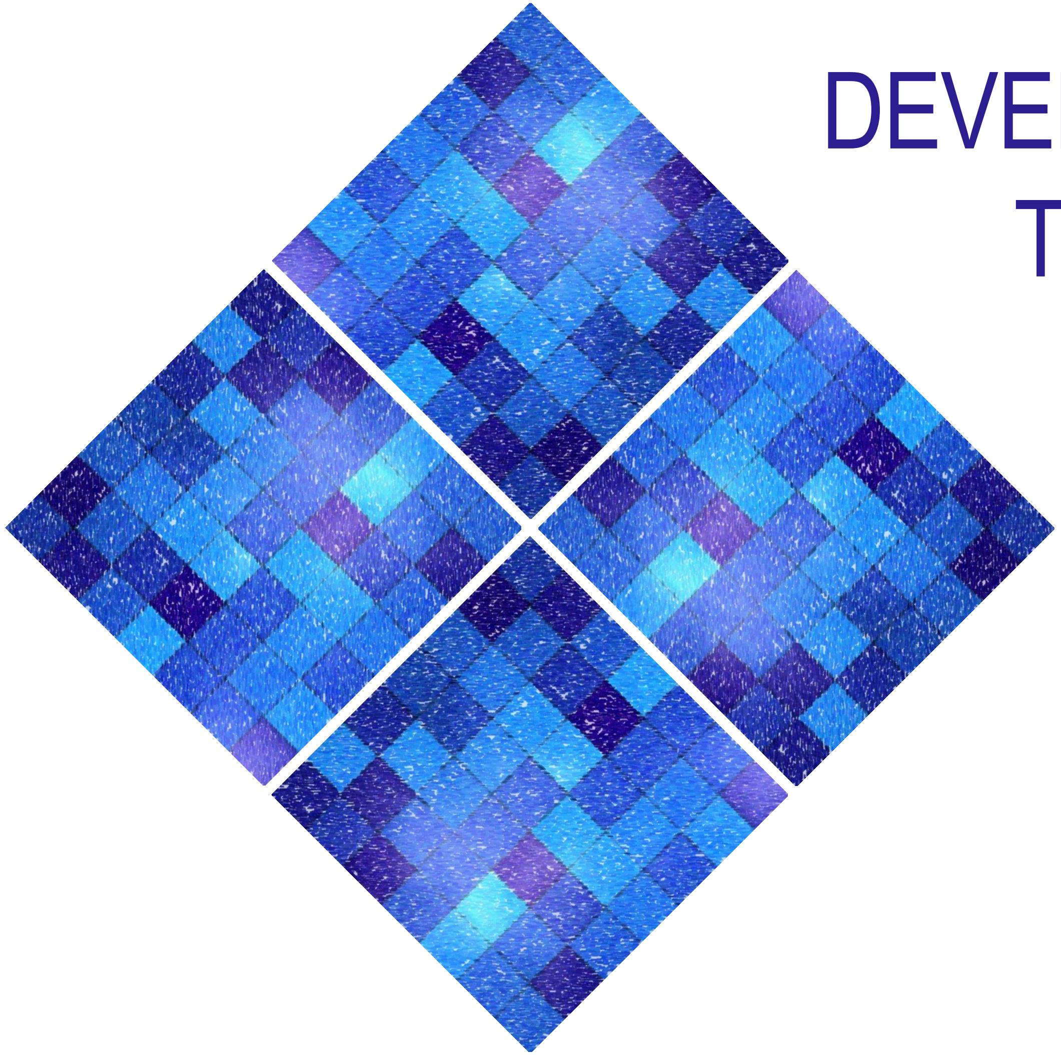


DEVELOPING SOFTWARE THAT UNDERSTANDS THE REAL WORLD

Dimitri Lyras,
Director of Lyras Shipping,
Founder of Ulysses Systems

Computers and the real world

- Software today is mostly geared towards passing commands to machine in the most expedient way. It’s not about teaching computers about the w they will be operating in

- However to prevent undesired behavior in computerized systems it is not enough for computers to perform routines; they have to make judgement situations

- To make such judgements they have to understand situations. We need means to teach computers situations

Command Oriented Input

- Currently, this is done in a way that is based on maximizing the efficiency of passing commands to computers.

- Passing commands to computers is not an efficient way to describe situa

- There are plenty of ways to teach computers the situations they will be working in and to ensure they behave desirably.

- However this is not applied widely today as we are still using technology enabling commands

Searching for Stored Information

Mr. President, what was your experience in prison after being arrested in 1962

```
SELECT e.ID, e.prisonerID, e.prisonID
FROM tblPRIexperience e
INNER JOIN tblPRIprisoner PR ON PR.ID = e.prisonerID
INNER JOIN tblPRISON P ON P.ID = e.prisonID
WHERE e.PrisonID IN ('D3A71B4C-52BF-40AA-A4B6-212C60DD88F8', '26A56662-B431-4E61-B95F-8F61BE6379B2', 'B87F5C5D-173B-45F0-B8D0-9BB44D2B775E')
AND e.PrisonerID = '66C6E52D-3975-4EDE-946E-50C6A2285BE5‘
AND e.StartDate > 1962
```

Comprehension of automation

- The huge gulf between a computer understanding the world it operates within, and a computer understanding commands has been bridged only minimal way

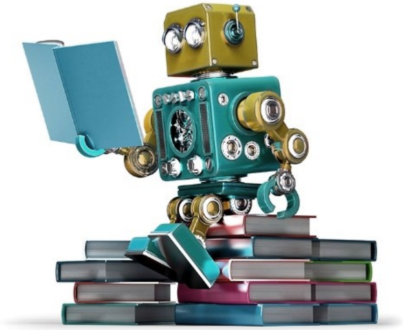
- Of course data models and rules exist that try to emulate the world. How the structure of data models and rules is primarily aimed at making it eas developers to pass commands to computers

- There has been allot less work put into providing structures for describin world; such as the context

Machine Learning

- So neither the designers nor the end users can get an efficient grasp of w the computer has been taught via the data models and rules

- Machine learning attempts some of this but in focusing on correlation anc lacking understanding of cause and effect, machine learning is just one s in the right direction. There are many more



When do you need modelling?

- Whenever you are designing something; architects, mechanical and electrical engineers always use models

- When you need to remember how you defined a set of relationships

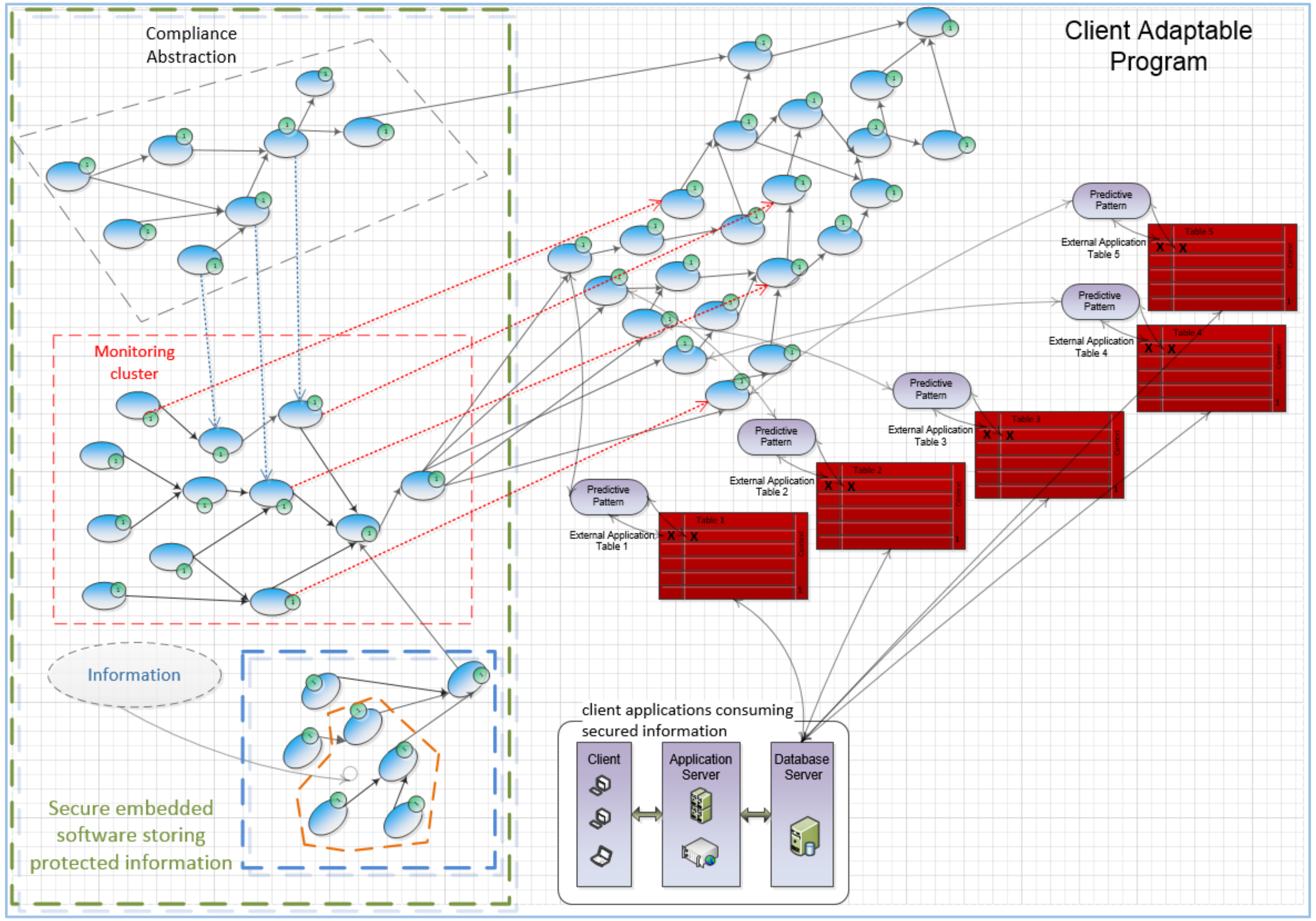
When do you need models most?

- When there are many exceptions to rules
- When there are variations in processes
- When there is a need to predict behavior in a variety of situations
- When complex logic needs to be extended
- When complex logic needs to be maintained
- When you need a lot of logic to be easily scrutinized by stakeholders

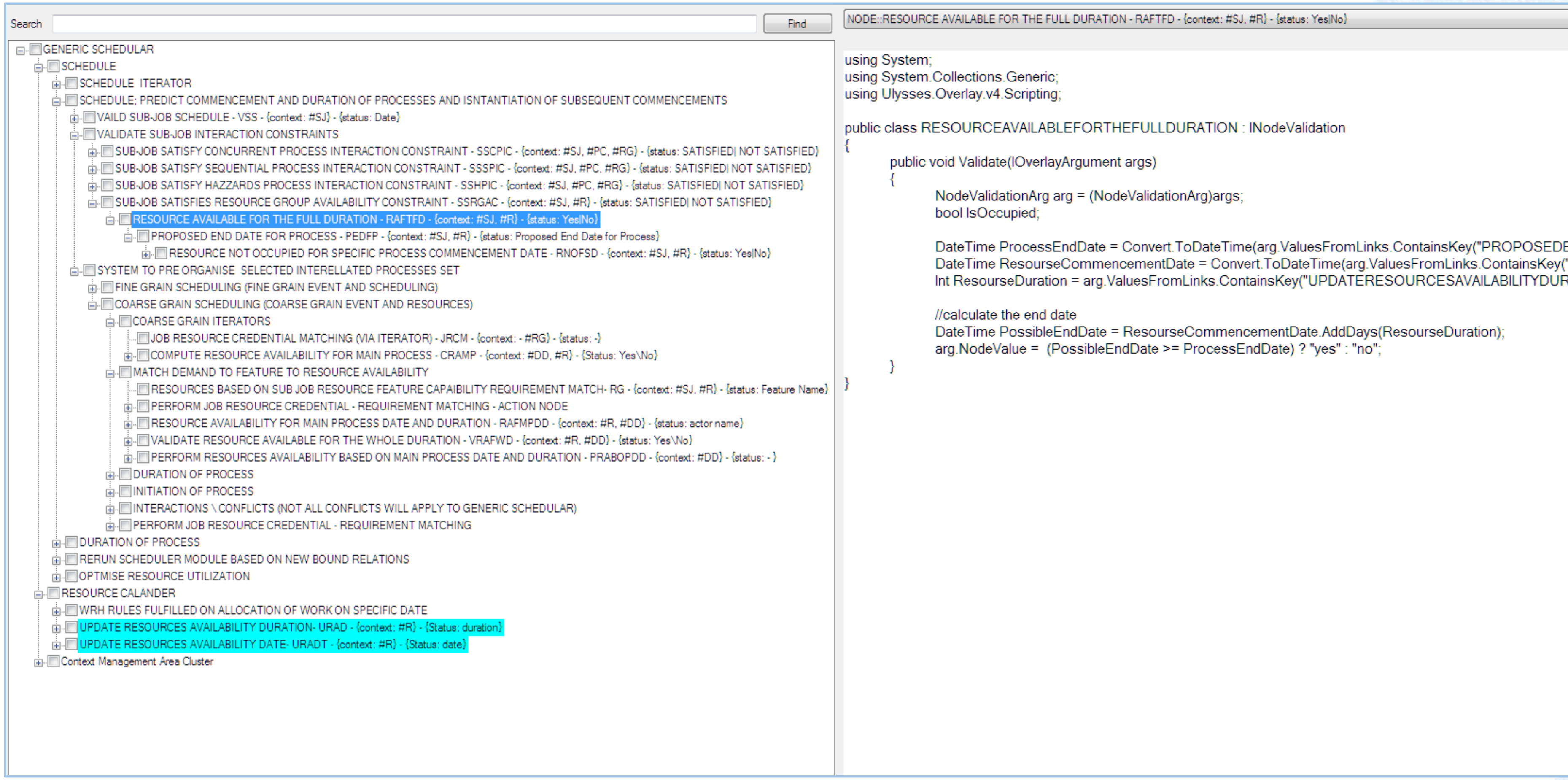
Example of logic needing to be scrutinized by many stakeholders

Imagine a self-driving car making decisions. How many stakeholders are affected?

Example: secure data needing to be protecte from unauthorized access:



Example scheduling repairs in the marine business



```
GENERIC SCHEDULAR
  SCHEDULE
    SCHEDULE  ITERATOR
    SCHEDULE; PREDICT COMMENCEMENT AND DURATION OF PROCESSES AND ISNTANTIATION OF SUBSEQUENT COMMENCEMENTS
      VAILD SUB-JOB SCHEDULE - VSS - {context: #SJ} - {status: Date}
      VALIDATE SUB-JOB INTERACTION CONSTRAINTS
        SUB-JOB SATISFY CONCURRENT PROCESS INTERACTION CONSTRAINT - SSCPIC - {context: #SJ, #PC, #RG} - {status: SATISFIED| NOT SATISFIED}
        SUB-JOB SATISFY SEQUENTIAL PROCESS INTERACTION CONSTRAINT - SSSPIC - {context: #SJ, #PC, #RG} - {status: SATISFIED| NOT SATISFIED}
        SUB-JOB SATISFY HAZZARDS PROCESS INTERACTION CONSTRAINT - SSHPIC - {context: #SJ, #PC, #RG} - {status: SATISFIED| NOT SATISFIED}
        SUB-JOB SATISFIES RESOURCE GROUP AVAILABILITY CONSTRAINT - SSRGAC - {context: #SJ, #R} - {status: SATISFIED| NOT SATISFIED}
          RESOURCE AVAILABLE FOR THE FULL DURATION - RAFTFD - {context: #SJ, #R} - {status: Yes|No}
            PROPOSED END DATE FOR PROCESS - PEDFP - {context: #SJ, #R} - {status: Proposed End Date for Process}
              RESOURCE NOT OCCUPIED FOR SPECIFIC PROCESS COMMENCEMENT DATE - RNOFSD - {context: #SJ, #R} - {status: Yes|No}
      SYSTEM TO PRE ORGANISE  SELECTED INTERELLATED PROCESSES SET
        FINE GRAIN SCHEDULING (FINE GRAIN EVENT AND SCHEDULING)
        COARSE GRAIN SCHEDULING (COARSE GRAIN EVENT AND RESOURCES)
          COARSE GRAIN ITERATORS
            JOB RESOURCE CREDENTIAL MATCHING (VIA ITERATOR) - JRCM - {context: - #RG} - {status: -}
            COMPUTE RESOURCE AVAILABILITY FOR MAIN PROCESS - CRAMP - {context: #DD, #R} - {Status: Yes\No}
          MATCH DEMAND TO FEATURE TO RESOURCE AVAILABILITY
            RESOURCES BASED ON SUB JOB RESOURCE FEATURE CAPAIBILITY REQUIREMENT MATCH- RG - {context: #SJ, #R} - {status: Feature Name}
            PERFORM JOB RESOURCE CREDENTIAL - REQUIREMENT MATCHING - ACTION NODE
            RESOURCE AVAILABILITY FOR MAIN PROCESS DATE AND DURATION - RAFMPDD - {context: #R, #DD} - {status: actor name}
            VALIDATE RESOURCE AVAILABLE FOR THE WHOLE DURATION - VRAFWD - {context: #R, #DD} - {status: Yes\No}
            PERFORM RESOURCES AVAILABILITY BASED ON MAIN PROCESS DATE AND DURATION - PRABOPDD - {context: #DD} - {status: - }
          DURATION OF PROCESS
          INITIATION OF PROCESS
          INTERACTIONS \ CONFLICTS (NOT ALL CONFLICTS WILL APPLY TO GENERIC SCHEDULAR)
          PERFORM JOB RESOURCE CREDENTIAL - REQUIREMENT MATCHING
    DURATION OF PROCESS
    RERUN SCHEDULER MODULE BASED ON NEW BOUND RELATIONS
    OPTMISE RESOURCE UTILIZATION
  RESOURCE CALANDER
    WRH RULES FULFILLED ON ALLOCATION OF WORK ON SPECIFIC DATE
    UPDATE RESOURCES AVAILABILITY DURATION- URAD - {context: #R} - {Status: duration}
    UPDATE RESOURCES AVAILABILITY DATE- URADT - {context: #R} - {Status: date}
  Context Management Area Cluster
```

NODE::RESOURCE AVAILABLE FOR THE FULL DURATION - RAFTFD - {context: #SJ, #R} - {status: Yes|No}

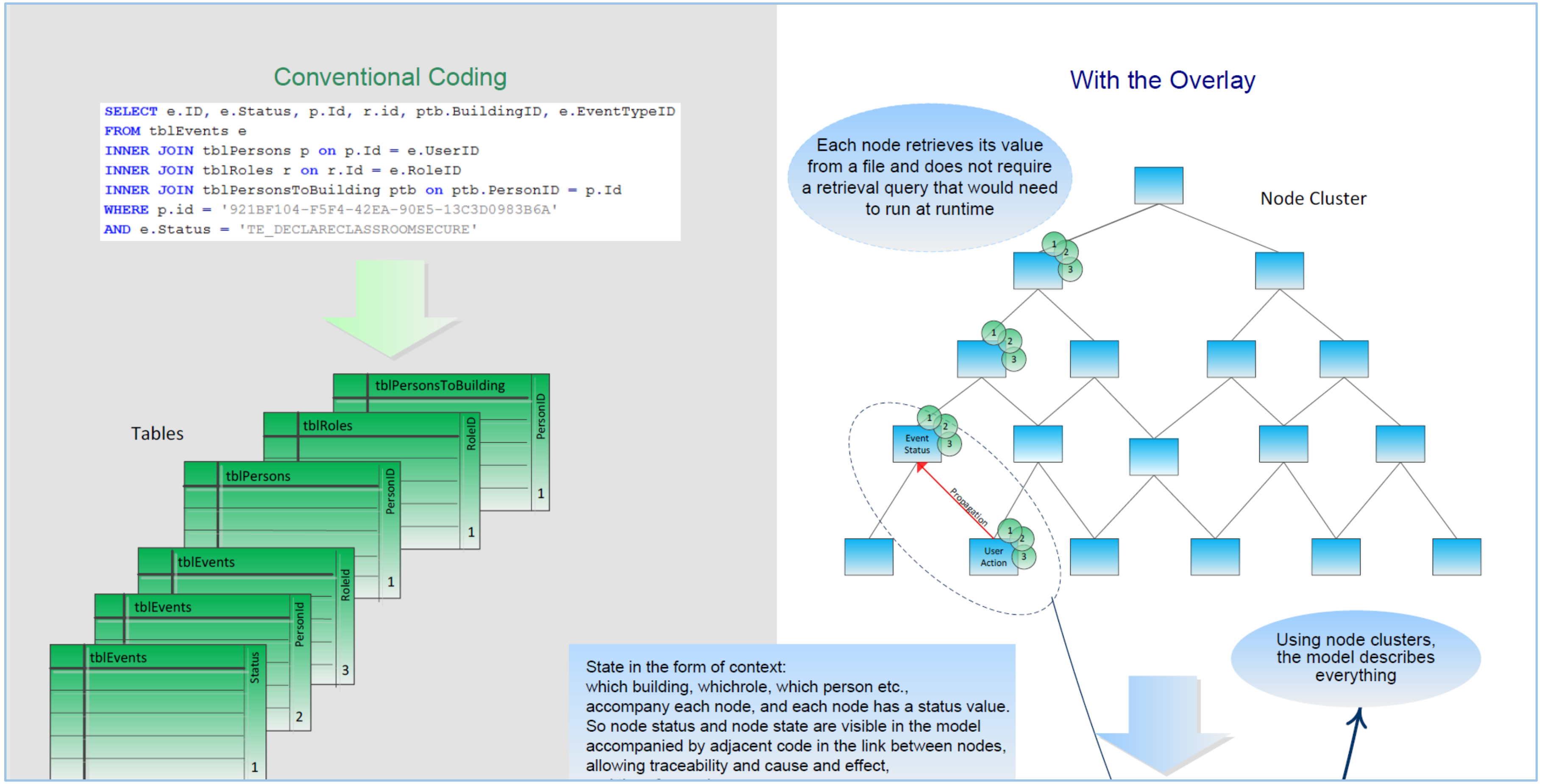
```
using System;
using System.Collections.Generic;
using Ulysses.Overlay.v4.Scripting;

public class RESOURCEAVAILABLEFORTHEFULLDURATION : INodeValidation
{
        public void Validate(IOverlayArgument args)
        {
                NodeValidationArg arg = (NodeValidationArg)args;
                bool IsOccupied;

                DateTime ProcessEndDate = Convert.ToDateTime(arg.ValuesFromLinks.ContainsKey("PROPOSEDE
                DateTime ResourseCommencementDate = Convert.ToDateTime(arg.ValuesFromLinks.ContainsKey("
                Int ResourseDuration = arg.ValuesFromLinks.ContainsKey("UPDATERESOURCESAVAILABILITYDUR

                //calculate the end date
                DateTime PossibleEndDate = ResourseCommencementDate.AddDays(ResourseDuration);
                arg.NodeValue =  (PossibleEndDate >= ProcessEndDate) ? "yes" : "no";
        }
}
```

Executing models versus non executing models



Conventional Coding

```sql
SELECT e.ID, e.Status, p.Id, r.id, ptb.BuildingID, e.EventTypeID
FROM tblEvents e
INNER JOIN tblPersons p on p.Id = e.UserID
INNER JOIN tblRoles r on r.Id = e.RoleID
INNER JOIN tblPersonsToBuilding ptb on ptb.PersonID = p.Id
WHERE p.id = '921BF104-F5F4-42EA-90E5-13C3D0983B6A'
AND e.Status = 'TE_DECLARECLASSROOMSECURE'
```

With the Overlay

Each node retrieves its value from a file and does not require a retrieval query that would need to run at runtime

State in the form of context: which building, whichrole, which person etc., accompany each node, and each node has a status value. So node status and node state are visible in the model accompanied by adjacent code in the link between nodes, allowing traceability and cause and effect,

Using node clusters, the model describes everything

THANK YOU

Since 1996, Ulysses Systems is a leader in software innovation for the Maritime Industry, providing management solutions to ship-owners and ship managers. Its award winning software, Task Assistant® enables both office and seagoing personnel to work intuitively, efficiently and effectively. Class certified by DNV, Bureau Veritas and ClassNK, Task Assistant® is designed to require minimal training. Managers should expect a fast return on total software lifecycle cost from reliable and mature process optimization and minimization of information gaps.

info@ulysses-systems.com

ATHENS - LONDON - MUMBAI - CHICAGO

www.ulysses-systems.com