The Conflict between Emergent Behaviours and Predictability in Distributed Computing

Rogério de Lemos
Computing Laboratory - University of Kent, UK

- Motivation
- Process and data representation
- Artificial immune systems
- Architectural exception handling
- Conclusions
Motivation: why are we here?

Existing distributed systems techniques do not **scale** for envisaged future systems;

- Decentralised peer-to-peer, self-organised ad hoc networks, ubiquitous systems, etc.

Traditional approaches specify system properties in a top-down manner;

- Amenable to rigorous or mathematical analysis;
  - key system/component variables that describe behaviour are well identified: inputs, outputs, internal and environmental states;
**Motivation: why are we here?**

How *decentralised control techniques* can be used in building distributed systems?

- Large population of entities that independently gather information and decide how to behave;
  - entities with simple functionality and limited communication;
  - macroscopic behaviours from microscopic behaviours and interactions;

- Emergent outcome would be to provide solutions to specific problems;
Motivation: why are we here?

Bridging the gap between two communities:

- distributed systems (consensus) and computational intelligent (emergence/learning);
  - **consensus**: protocols to reach agreement under particular failure loads;
  - **emergence**: a collection of simple autonomous and self-sufficient entities that are able to adapt to changing environments;
  - **learning**: refers to the automatic mining of information from available data to create knowledge;
Motivation: why am I here?

First contact with computational intelligence: how to build safety cases for industrial controllers using artificial neural networks?

- How to construct a logical argument showing that system safety would not be breached?
  - the argument relies on environmental conditions, inputs to the system, its state, or any additional protection or warning devices;
  - demonstrate that bad behaviours seldom occur, if they do, the risk is negligible;
    - evidence provided from forward and backward analysis;
Motivation: why am I here?

- Problems associated with computational intelligence based solutions have been (temporally) “solved”:
  - mission critical systems do not allow any kind of adaptability (which can be a source of uncertainty);

Due to pressures of system complexities, solutions are being investigated that isolate the uncertainties associated with non-conventional solutions:

- transport management (air, rail, and road);
  - safety in numbers does not apply to transportation!!!
Motivation: why am I here?

Immunised fault tolerance for mechatronic devices:
- adaptive error detectors;

Software engineering applied to large multi-agent systems:
- from the dependability perspective;

Game theory in multi-criteria decision problems:
- selection of software components;
  - more appropriate to Nash equilibrium rather game playing, e.g., bargain games;
Motivation: how to cope with change?

Bio-inspired Computing:

- Exploit clever mechanisms from nature by defining algorithms and implementations that are appropriate for the problem at hand;
  - The “imperfect metaphor”: use the relevant bits, and ignore the rest;
- Inspiration should be based on logical characteristics rather than physical:
  - these tend to be more generic and more appropriate to analyse data in a virtual data space;
Motivation: how to cope with change?

Control Systems:

- Observability and controlability of a system variables allow its behaviour to be controlled:
  - feedback control loops to control the system dynamics:
    - monitoring, diagnosis, adaptation, learning, and interpretation;

Social and Economic Models:

- Evolutionary game theory:
  - interactions among a large population of entities, whose decisions are simple and selfish;
Motivation: how to cope with change?

Self-* Approaches:

Self-capabilities identify complementary mechanisms that can coexist in the same systems;

- **Self-managing** systems have the ability to react and dynamically adapt to changes in their environment;
- **Self-configuring** systems are able to modify the interactions among components;
- **Self-healing** systems can dynamically reveal and fix their own bugs.
- **Self-optimising, self-organising, and self-protecting** ;
Process and Data Descriptions

Simon has identified two ways of describing complex systems for their understanding: *process* and *state* (data [MacFarlane]):

- **Data descriptions:**
  - “A circle is the locus of all points equidistant from a given point”;
  - Characterises the world as sensed;
    - Provide the criteria for identifying objects, often by modelling the objects themselves;
  - Pictures, blueprints, most diagrams, and chemical structural formulas;
  - E.g., representation of a circle as a bit map;
**Process and Data Descriptions**

- **Process descriptions:**
  - “To construct a circle, rotate a compass with one arm fixed until the other arm has returned to its starting point”;
  - If this process is followed, one gets a circle;
  - Characterises the world as acted upon;
    - Provide the means for producing or generating objects having desired characteristics;
  - Recipes, differential equations, and equations for chemical reactions;
  - E.g., representation of a circle as an equation;
Process and Data Descriptions

The major difference between these descriptions are the amount of data required for modelling:

- Process descriptions require less information;
  - data descriptions of systems might be unbounded;
Living organisms must develop correlations between goals in the sensed world and actions in the world of process;

- Given a desired state and a given state, the task is to find the correlating process between these two states;

- Problem solving in science:
  - Given the description of some phenomena, find the equations that will produce the phenomena;
Emergence vs. Predictable Behaviours

Data representations might be accurate but not precise:

- They are an abstraction of the actual behaviour;
- Data undergoes generalizations before it becomes meaningful;
- These two issues lead to uncertainties;
- Based on these uncertainties *emergent behaviours* are able to materialize;
Emergence vs. Predictable Behaviours

Process descriptions are precise but not accurate:

- The assumptions that allow a process to be realizable introduce uncertainties;
- Some of the assumptions are discharged to obtain more accurate models;
- The lack of uncertainties determine the *predictable behaviours*;
Adaptability and Predictability

- Anomaly detection using AIS techniques:
  - Data oriented solution;
  - Generate adaptable error detectors for unexpected and undesirable circumstances:
    - We have noticed a decrease in the detection accuracy;
  - Exceptional behaviour at the architectural level;
    - Process oriented solution;
    - Build adaptable, but deterministic systems:
      - Uncertainties are eliminated from the system behaviour;
      - Application oriented solutions that difficult to scale;
  - Insufficient when dealing with uncertainties of attacks;
Why the Biology and the Immune System?

Biology seems to be good at solving problems!

- To be inspired by nature but do not copy it blindly;

In the area of computational intelligence:

- Evolutionary algorithms, artificial neural networks, etc.
- Immune system has not been investigated a great deal;

What does the immune system offer?

- Pattern recognition, learning and memory, robust and distributed, adaptive, and diverse;
The immune system is a defence mechanism against body invaders (viruses, bacteria, and fungi):

- Four layers of defence: physical barriers, chemical barriers, innate immunity, and adaptive immunity;

The main features of the adaptive immunity:

- Defence against specific foreign organisms, and its ability to learn and remember;

Three main theories describe the immune system:

- Clonal selection, immune network, and danger;
Adaptive immunity from the clonal selection perspective:

- Train immature lymphocyte cells to distinguish between self molecules and foreign antigens;
  - Immune response is undertaken by mature lymphocyte cells;
- During immune response, mature lymphocyte cells multiply for effective defence, and differentiate for perfect recognition;
- After immune response, efficient mature lymphocyte cells are stored for subsequent encounters;
Immune System

Immune system main features:

- Pattern recognition (anomaly detection, noise tolerance), diversity, learning, memory, redundancy, robustness, feature extraction, distributed, multi-layered, and adaptive;
Artificial Immune Systems (AIS)

Artificial immune system (AIS) is an example of nature-inspired problem solving system.

- Inspired in the current understanding of the mammal immune system,
  - it does not follow exactly its biological steps;

Artificial immune system (AIS):

- AIS are computational systems inspired by theoretical immunology and observed immune functions, principles and models, which are applied to complex problem domains;
Framework for Artificial Immune Systems (AIS)

- **Representation** of antibodies and antigens:
  - numerical (discrete or continuous), or categorical;

- **Affinity measures** establish the similarity between an antibody and antigen;
  - e.g., Hamming, or Euclidean distance;

- **Immune algorithms** define the system dynamics;
Artificial Immune Systems (AIS)

AIS scope:

- fault and anomaly detection, security of information systems, data analysis (data mining, machine learning, pattern recognition), scheduling, autonomous control, optimisation, and robotics;

Some AIS techniques:

- Negative selection algorithms, immune network models, bone marrow models, clonal selection algorithms, danger theory, etc.
Negative Selection Algorithm

- Idea taken from the negative selection of T-cells in the thymus;
  - T-cells that match self are eliminated;
  - mature T-cells will in general match only non-self;
- Self/ non-self discrimination metaphor provided inspiration for the negative selection algorithm;
  - generate detectors to detect undesirable changes to normal patterns or behaviour (self) of a system;
  - applied initially to change-detection in computer security;
Negative Selection Algorithm

Split in two phases:

- **Censoring (“training”) phase:**
  - Generate random strings (Ro)
  - Match?
    - yes
    - no
      - Self strings (S)
      - Reject

- **Monitoring (“testing”) phase:**
  - Data set
  - Match?
    - yes
    - no
      - Detector set (R)
      - No detection of nonself
      - Nonself detected
Avizienis was the first to express the analogy between the immune system and fault tolerance:

- Autonomy, distributed lymphatic vessels among the body, exclusive communication links, redundancy, and diversity;

An immune system metaphor for error detection:

- self / non-self discrimination;

<table>
<thead>
<tr>
<th>Immune system</th>
<th>Fault Tolerance Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self molecules</td>
<td>Self states/behaviours</td>
</tr>
<tr>
<td>Nonself molecules (antigenic patterns)</td>
<td>Abnormal states/behaviours</td>
</tr>
<tr>
<td>T-cells</td>
<td>Error detectors</td>
</tr>
</tbody>
</table>
AIS in Anomaly Detection

Problems with negative selection in the classification of self/non-self:

- Generation of random detectors;
- Algorithms inefficient and time consuming (random search);
  - It does not use information from the self;
- In the training phase only negative examples (self) are used;
- Non-adaptive;
Other issues:

- What is self/non-self should depend on the size of erroneous data compared with non-erroneous;

- Discrete representations are problematic:
  - definition of affinity is not clear;
  - small mutations might lead to a great impact in the meaning of data;

- Minimise size of detector set and maximise coverage:
  - increase run-time efficiency;
Adaptable Error Detectors

Detectors that are able to adapt to changes in the environment, or replacement of components:

- An AIS system in which antibodies can be used across systems;

Immune network theory:

- Self-assertion view:
  - no initial knowledge of self, which is developed over time;
  - adaptability;

- Metadynamics:
  - recruitment of new individuals to the network structure;
Data Descriptions in Anomaly Detection

- How to learn from rare events?
  - No correlation between rare events;
  - Difficult to learn from known non-erroneous behaviours;

- How to improve coverage and reduce false positives?
  - Data is an abstraction of the actual behaviour;
  - Trade off between specialization and generalisation;
    - Combine diverse data oriented approaches:

- A problem in data oriented systems is the data itself:
  - Data has to be representative of the actual system;
  - We need to have a deep understanding of the data;
Exceptional Behaviour at the Architectural Level

iC2C_top

NormalActivity

upper_detector

COTS

lower_detector

iC2C_internal

AbnormalActivity

upper_detector
detector_top

Error Detector (1)

Error Detector (n)
detector_bottom

AbnormalActivity

Error Handler (1)

Error Handler (n)
iC2C_top

Error Diagnosis

iC2C_top

iC2C_bottom
Exceptional Behaviour at the Architectural Level

UML sequence diagram for a PIDTimeout exception;

:Boiler Controller
Normal Activity

:BC
Error Handler

:AFC
Error Handler

:afcldBOTTOM

:PID3 Controller
(COTS)

setConfiguration()

FailureException
("PIDTimeout")

FailureException
("NoResponse")

shutDown()

timeStep()

timeStep()

timeStep()

setConfiguration()

(stopped)
Emergence vs. Predictable Behaviours

Bio-inspired solutions are based on data representations:

- The model is based on a sample of data associated with the system;
- Learning capabilities might be able with some data uncertainties:
  - However, introduces another level of uncertainty;
- Emergent behaviours useful to deal with unexpected circumstances:
  - However, the system might become unpredictable;
Emergence vs. Predictable Behaviours

Architectural solutions are based on process representations:

- Architectural configuration relies on feedback control loop:
  - Monitor and control meta-parameters of behaviour, structure and environment;

- If behaviour is not predictable, reaction to change would be non-deterministic;

- Predictability is achieved by removing all operational uncertainties:
  - Require complete state specifications, otherwise unexpected circumstances can lead to failures;
Most of the work has focused on emergent behaviours, or functional properties;

- What about non-functional properties?
  - e.g., performance, usability, and dependability;
  - the composition of two reliable system does not necessarily lead to a reliable system;

- Can a dependable system be obtained as a result of a collection of individual undependable entities?
  - How dependability can become an emergent global system property?
Emergent QoS

On the positive note: we might be able to build robust systems, but not robust entities:

- The notion of what is a system might have to change;
  - e.g., a population of PDAs where the notion of an individual becomes irrelevant;

- The sense of ownership/stakeholder is lost from the individual perspective:
  - BA cancels 1000 flights for the benefit of the rest of its passengers;
According to H.A. Simon there are two ways for representing systems:

- **Data descriptions** are accurate but imprecise;
  - abstractions and generalizations lead to uncertainties;
  - e.g., adaptable error detection using an AIS solution;

- **Process descriptions** are precise but inaccurate;
  - uncertainties are removed by discharging assumptions;
  - e.g., exceptional handling at the architectural level;

Uncertainties lead to emergent behaviours;
Promising Solutions?

Instead of discrete models, should not the problem be investigated from a continuous perspective?
  - The solution might be on the numbers;

What about “development biology”?
  - Modelling the process of growing;
    - How natural systems grow from a single seed?
  - E.g., self-repair would be an emergent property;
    - However, what about the failure assumptions?
Questions?

Thanks to:
- Jon Timmis, Modupe Ayara;
- Cecília Rubira, Alexandre Romanovsky,
  Paulo Asterio de C. Guerra