Leveraging Machine Learning models for Complex System Architecture & Design Decisions

TUTORIAL

Dr. Ramakrishnan Raman, ESEP
Principal Systems Engineer, Honeywell
Asst. Director – Asia Oceania, INCOSE
ramakrishnan.raman@incose.net
https://www.linkedin.com/in/ramakrishnanraman/

IARIA

The Sixteenth International Conference on Systems
ICONS 2021
April 18, 2021 to April 22, 2021 - Porto, Portugal

Dr. Ramakrishnan Raman

Expert Systems Engineering Professional (ESEP),

INCOSE Outstanding Service Award Recipient

Asst Director – INCOSE Asia Oceania, Senior Member - IEEE

Education & Certifications

- B.Tech (1995), MS (1997) – IIT Madras
- PhD (2019) – IIIT Bangalore
- Honeywell Six Sigma Plus Black Belt, 2003
- General Management Program - IIM Bangalore, 2005
- INCOSE Certified Systems Engineering Professional (CSEP), 2005-17
- MBA - ICFAI University, 2012
- INCOSE Certified Expert Systems Engineering Professional (ESEP), 2018
- Machine Learning certification courses, including Reinforcement Learning

Areas of Expertise

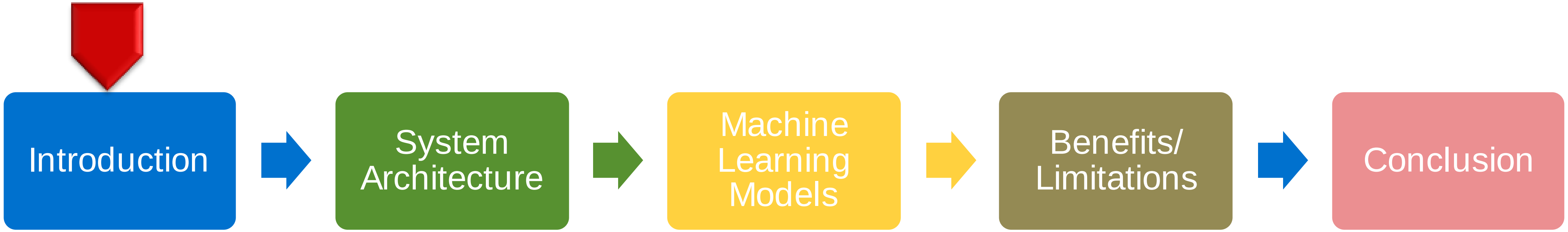
- Systems Engineering – Complex Systems, System-of-Systems, Model Based Systems Engineering, System Architecture & Design
- Artificial Intelligence – Machine Learning, Reinforcement Learning
- Software Architecture & Design; OOAD (Object-Oriented Analysis & Design) & Design Patterns
- RTCA/DO standards for Avionics Software development, SAE ARP 4754/ 4761
- Redundancy Architectures & Fault Tolerance, Distributed Systems

Tutorial Outline

Introduction → System Architecture → Machine Learning Models → Benefits/ Limitations → Conclusion

[This tutorial material is not a publication. The material is compiled from various sources for academic/ teaching purposes]

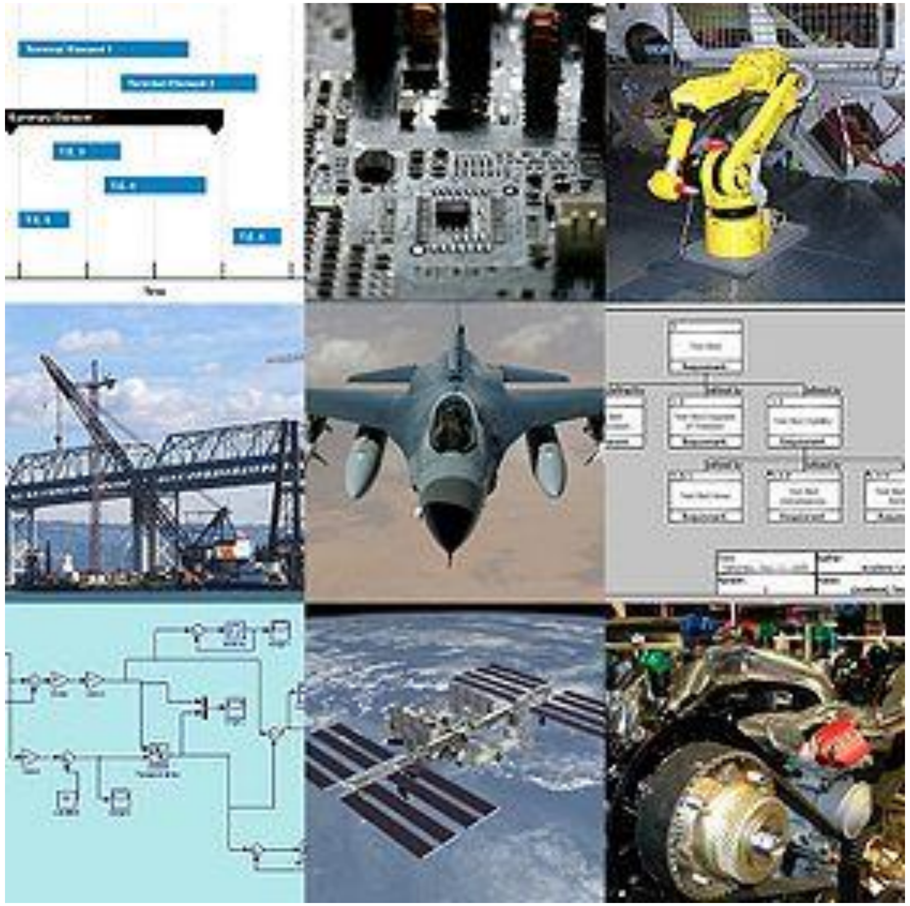
Introduction



- Systems
- System-of-Systems

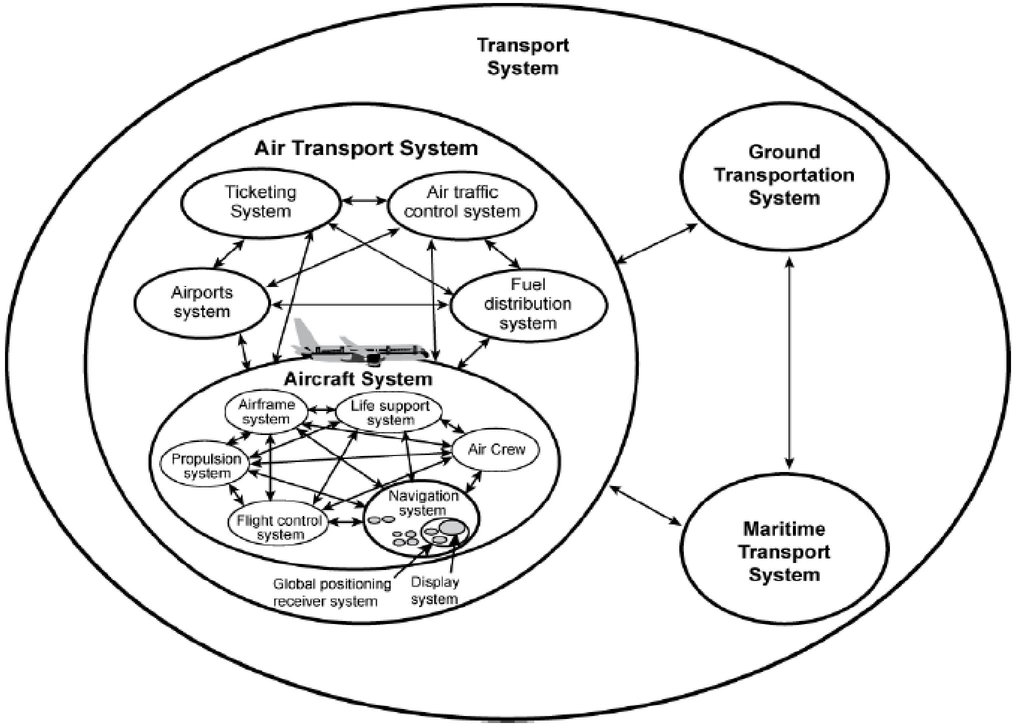
System

- A purposeful collection of inter-related components working together towards some common objective.
- A system may include software, mechanical, electrical and electronic hardware and be operated by people.
- System components are dependent on other system components
- The properties and behavior of system components are inextricably inter-mingled
- Systems are man-made, created and utilized to provide services in defined environments for the benefit of users and other stakeholders



System-of-Systems (SoS)

- System-of-Systems are systems-of-interest whose system elements are themselves systems - they typically entail large-scale inter-disciplinary problems involving multiple, heterogeneous and distributed systems
- Each system has an independent purpose and viability, in addition to the SoS by itself having an independent purpose and viability
- Typically entail large scale interdisciplinary problems involving multiple, heterogeneous, distributed systems



Source: INCOSE SE Handbook

What is Systems Engineering?

Systems Engineering *is a* ***transdisciplinary*** *and* ***integrative*** *approach to enable the successful realization, use, and retirement of* ***engineered systems****, using* ***systems principles and concepts****, and scientific, technological, and management methods.*

We use the terms "engineering" and "engineered" in their widest sense: "the action of working artfully to bring something about". "Engineered systems" may be composed of any or all of people, products, services, information, processes, and natural elements.

Engineered System Definition

An **engineered system** *is a system designed or adapted to interact with an anticipated operational environment to achieve one or more intended purposes while complying with applicable constraints.*

Thus, an "engineered system" is a system – not necessarily a technological one – which has been or will be "systems engineered" for a purpose.

Most General "System" Definition

A **system** *is an arrangement of parts or elements that together exhibit behaviour or meaning that the individual constituents do not.*

Systems can be either physical or conceptual, or a combination of both.

Systems in the physical universe are composed of matter and energy, may embody information encoded in matter-energy carriers, and exhibit observable behaviour.

Conceptual systems are abstract systems of pure information, and do not directly exhibit behaviour, but exhibit "meaning". In both cases, the system's properties (as a whole) result, or emerge from:

- the parts or elements and their individual properties; AND
- the relationships and interactions between and among the parts, the system and its environment.

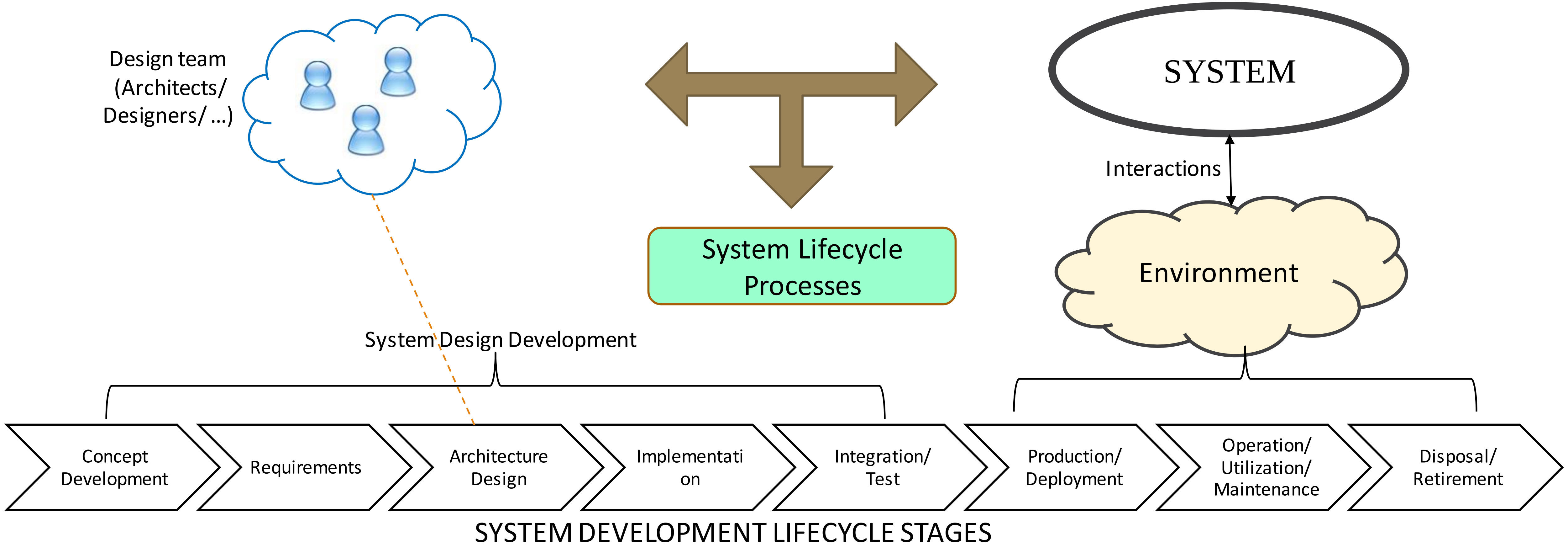
Definitions of the International Council on Systems Engineering (INCOSE) 2019

System Architecture

Introduction → System Architecture → Machine Learning Models → Benefits/ Limitations → Conclusion

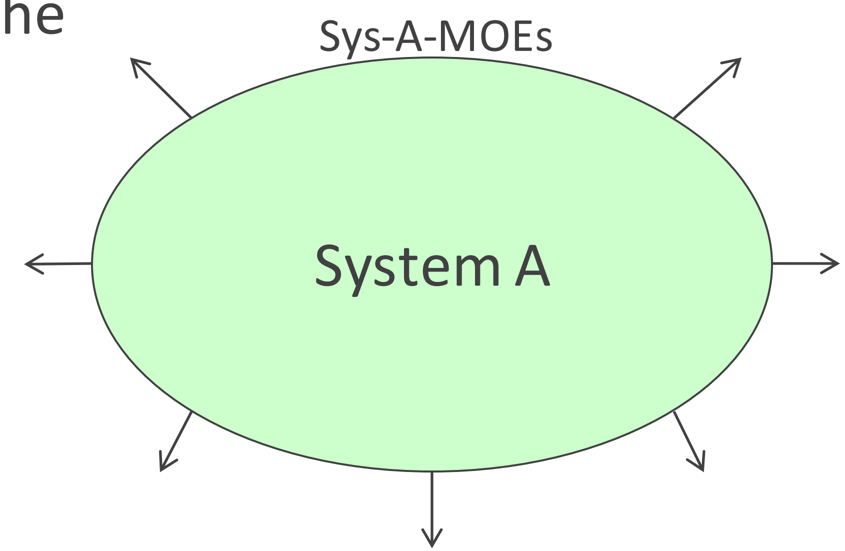
- MOEs
- System Architecture
- Arch Decisions
- Complex Systems
- Uncertainty

System Lifecycle Stages



Measures of Effectiveness - MOEs

- Operational measures of success that are closely related to the achievement of the objective of the system of interest
- Related to the achievement of the mission or operational objective being evaluated
  - In the intended operational environment
  - Under a specified set of conditions
- Manifest at the boundary of the system
- Examples
  - Response time to a user action
  - Time to Alert
  - Availability of the system



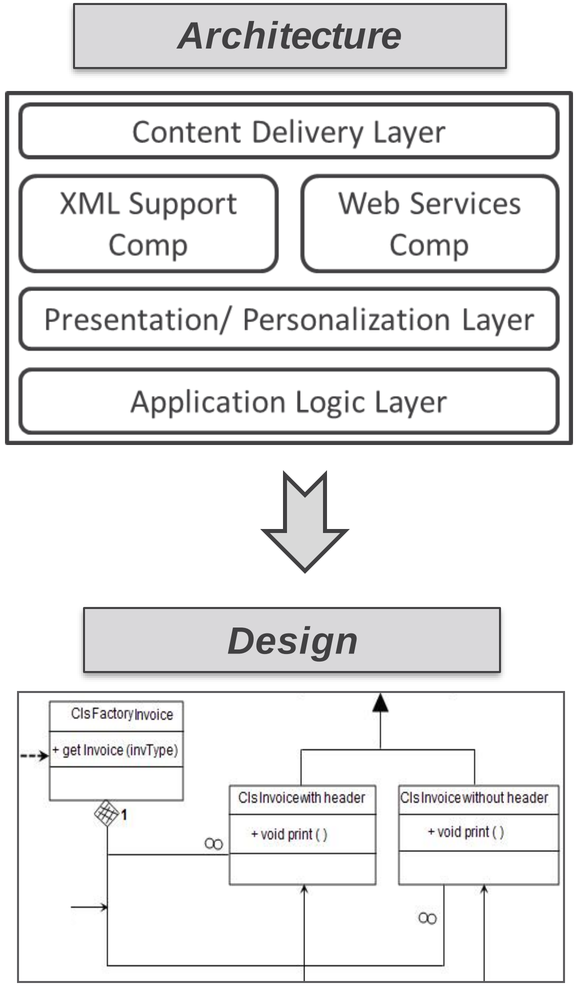
Each system is associated with a desired set of MOEs.

System Architecture

"...Arrangement, theme, and principles behind the various subsystems/ elements and their interactions to meet the system requirements and non-functional/ quality attributes.."

".. Architecture is the fundamental concepts/ properties of a system in its environment, embodied in its elements, relationships, and in the principles of its design and evolution

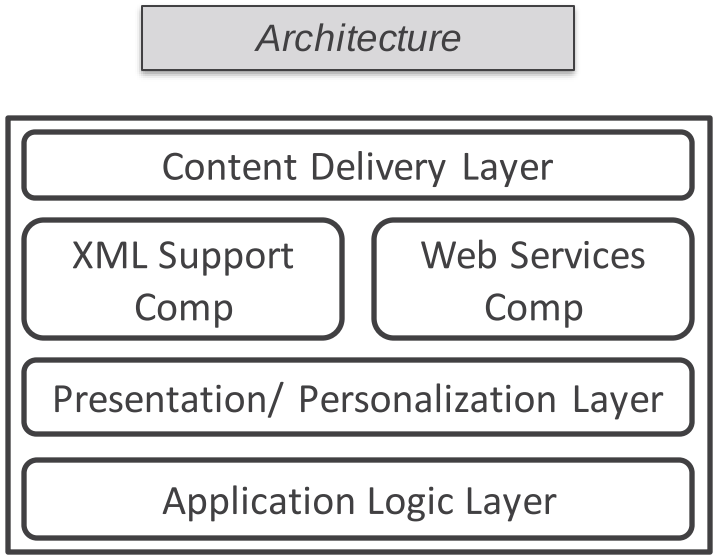
"...The quality and longevity of the system is largely determined by its architecture..."



Architecture Definition

ISO/IEC/IEEE 15288: "….the purpose of architecture definition is to generate system architecture alternatives, to select one or more alternative(s) that frame stakeholder concerns and meet system requirements, and to express this in a set of consistent views.."

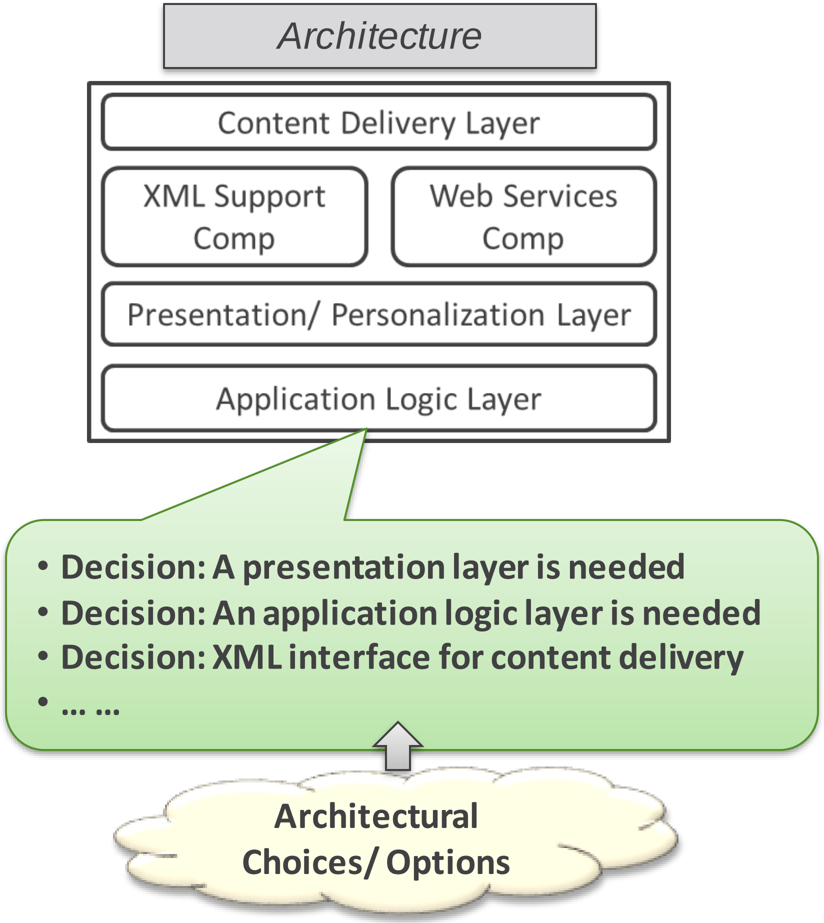
MFESA: System Architecture is the set of all of the most important, pervasive, higher-level, strategic <u>decisions</u>, inventions, engineering trade-offs, assumptions, and their associated rationales concerning how the system meets its allocated and derived product and process requirements.."



Architecting – "...a string of decisions..."

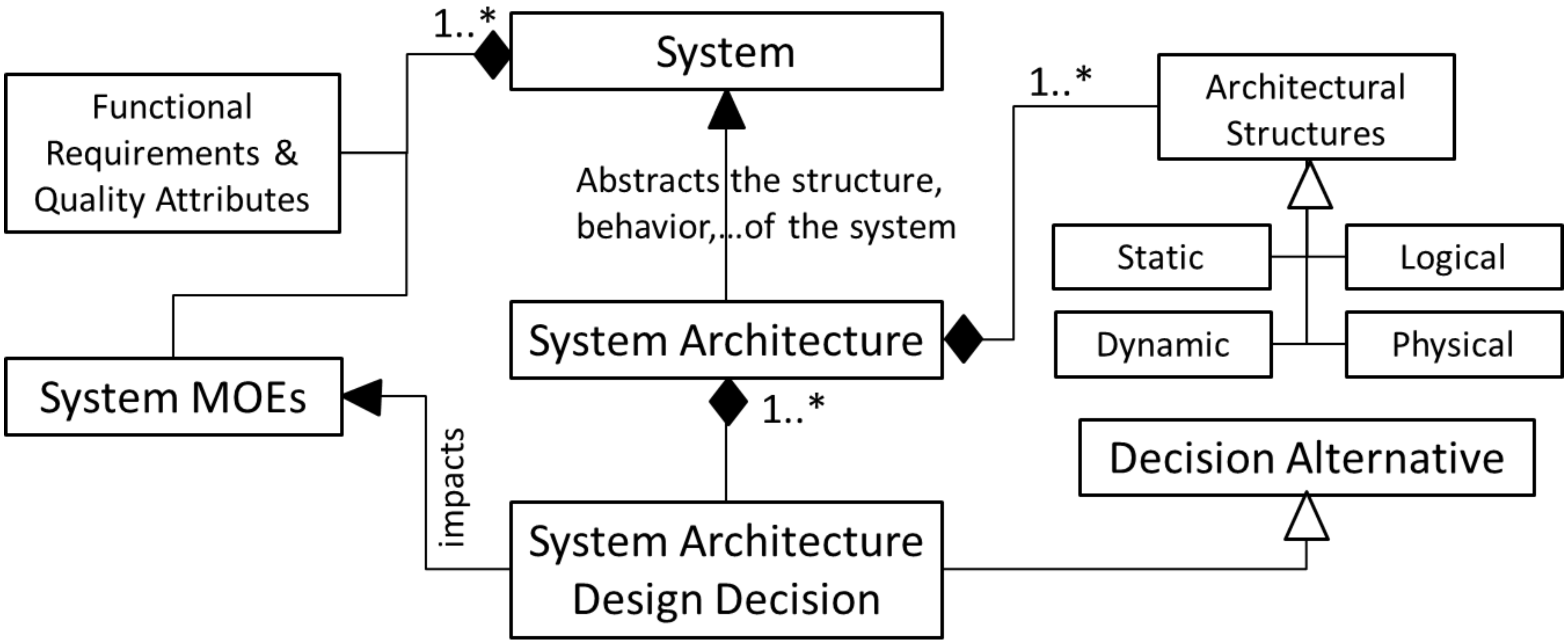
Arriving at the architecture can be viewed as a string of decisions to be taken, with each decision having one or more alternatives

Decision making techniques typically involve evaluating the alternatives in terms of how well each meet the requirements, thereby requiring tradeoffs



The available choices/ options need to be analyzed so as to meet the MOEs of the system

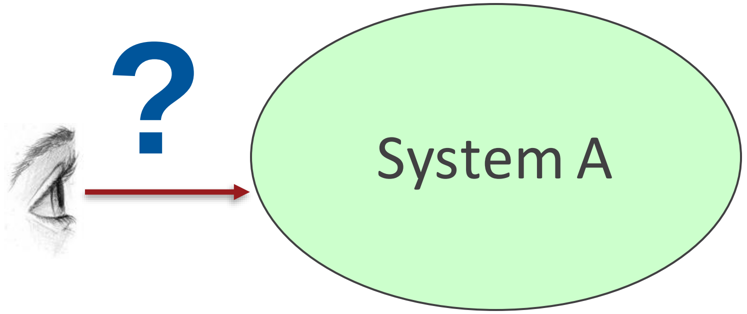
System Architecture



Complex Systems

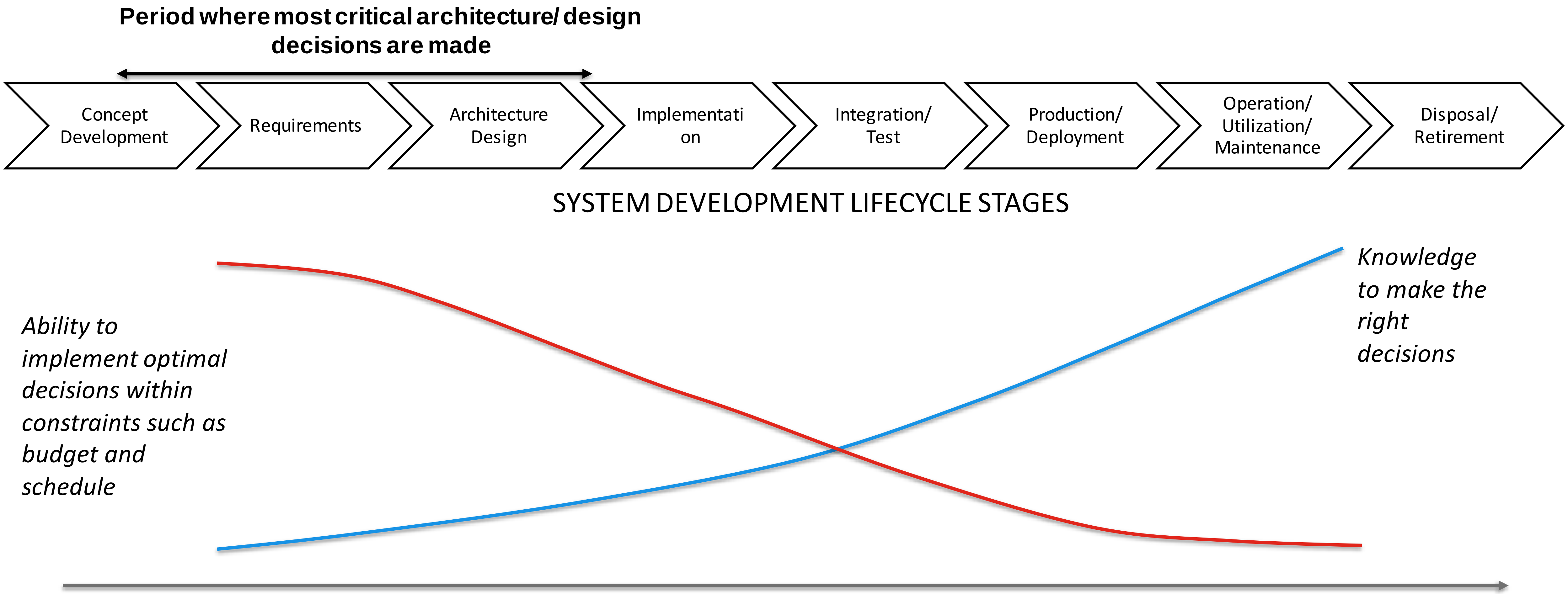
Complexity: Degree of difficulty in accurately predicting the future behavior

Complexity is determined by the system being observed, the capabilities of the observer, and the behavior that the observer is attempting to predict

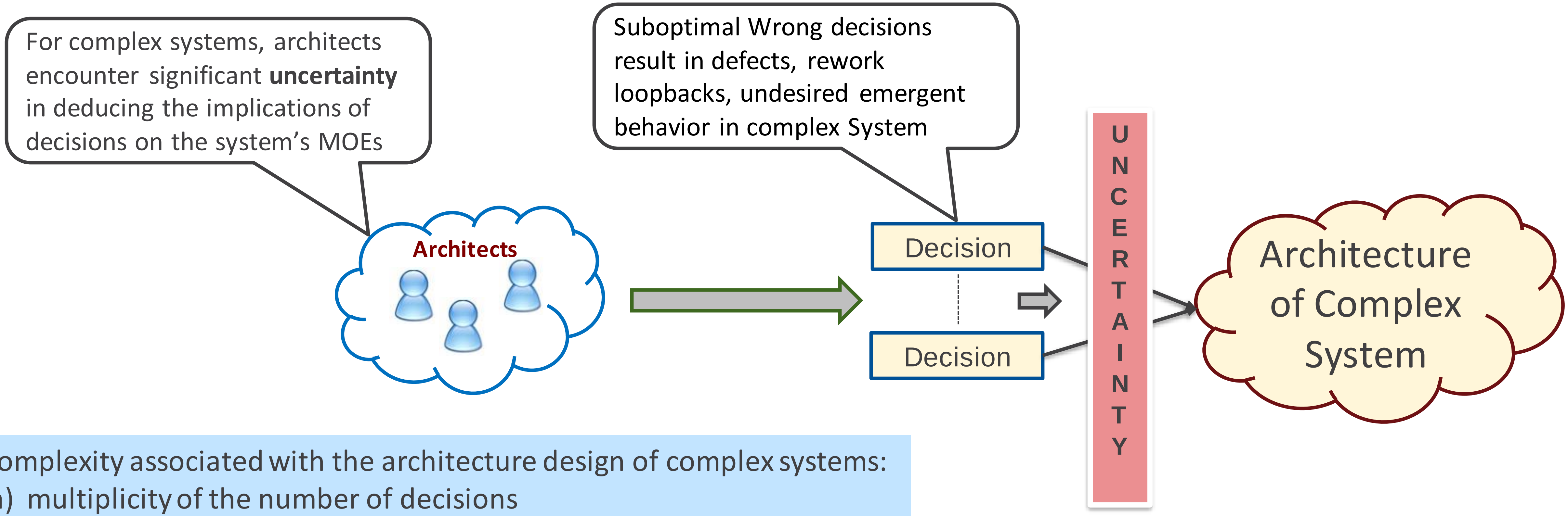


- Multiplex of relationships/ forces/ interactions between subsystems & constituent systems
- Difficulties in establishing cause-and-effect chain
- Characteristics: Emergence, hierarchical organization, numerosity….

Challenges in Design of Complex System



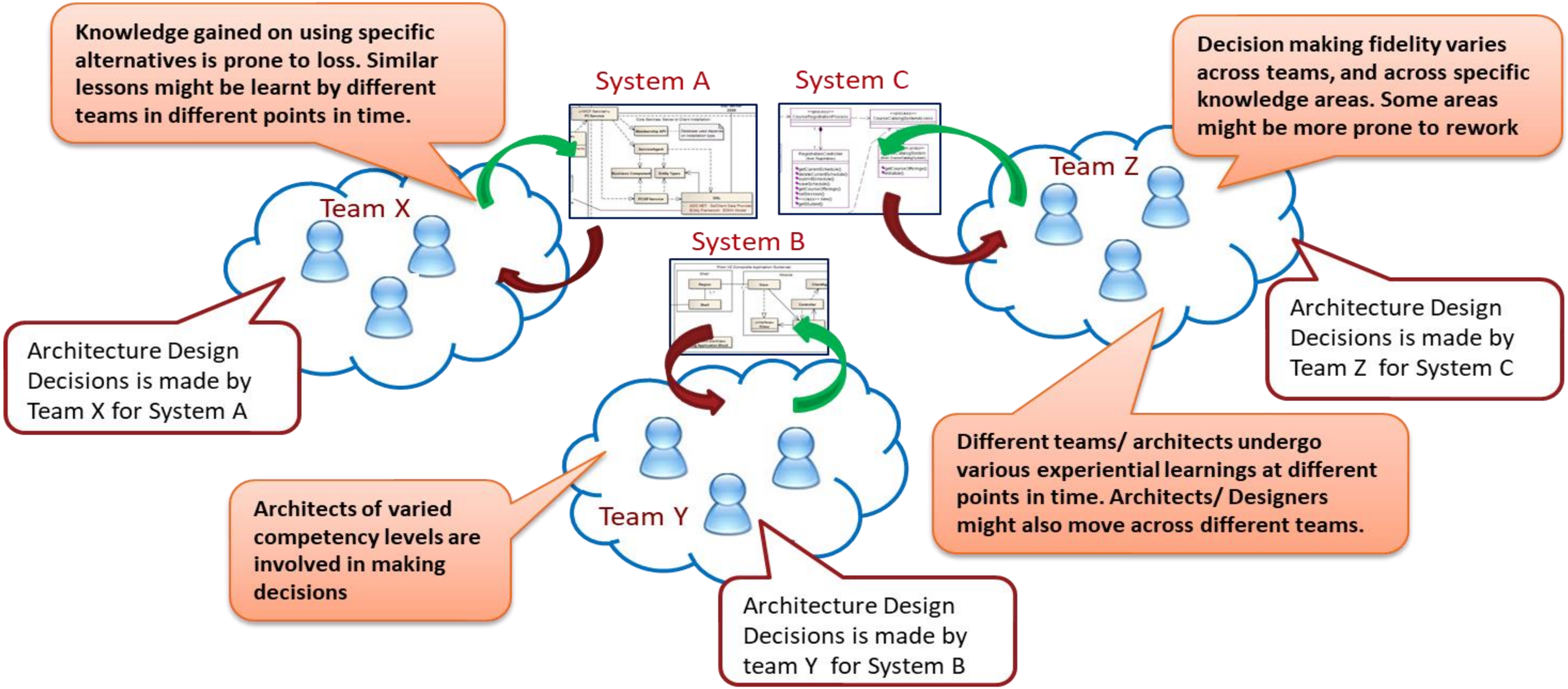
Uncertainty



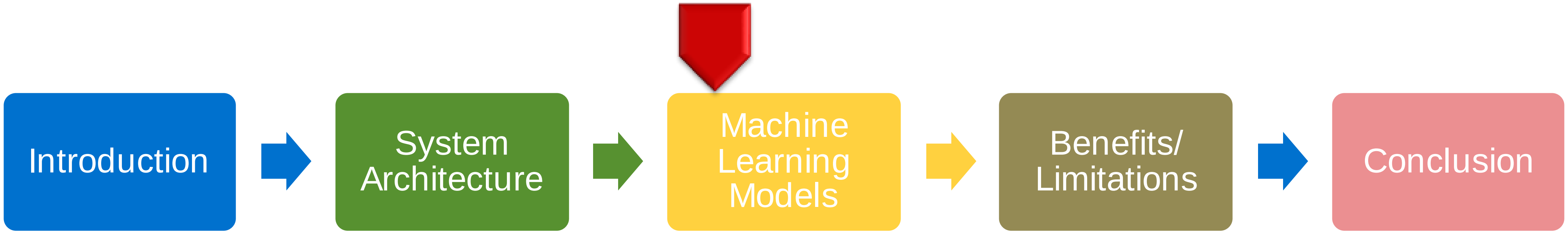
Complexity associated with the architecture design of complex systems:

(a) multiplicity of the number of decisions
(b) diversity of the knowledge areas pertaining to the decision
(c) Significant interdependencies and multiple implications of the decision

Organizational scenarios: architecting systems



System Architecture



- Machine Learning
- Decision Learning Cycles
- Decision Codification
- Learning & Prediction

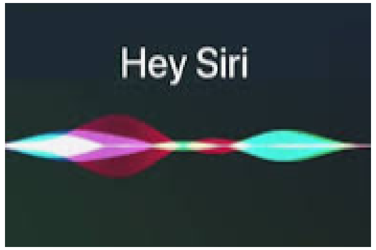
Machine Learning

Mohri M, Rostamizadeh A, Talwalkar A. Foundations of Machine Learning Cambridge, MA: MIT Press; 2012.

"Machine learning can be broadly defined as computational methods using experience to improve performance or to make accurate predictions"

"Machine Learning represents the field of study that allows computer programs to learn without being explicitly programmed"

Data + Output = Program

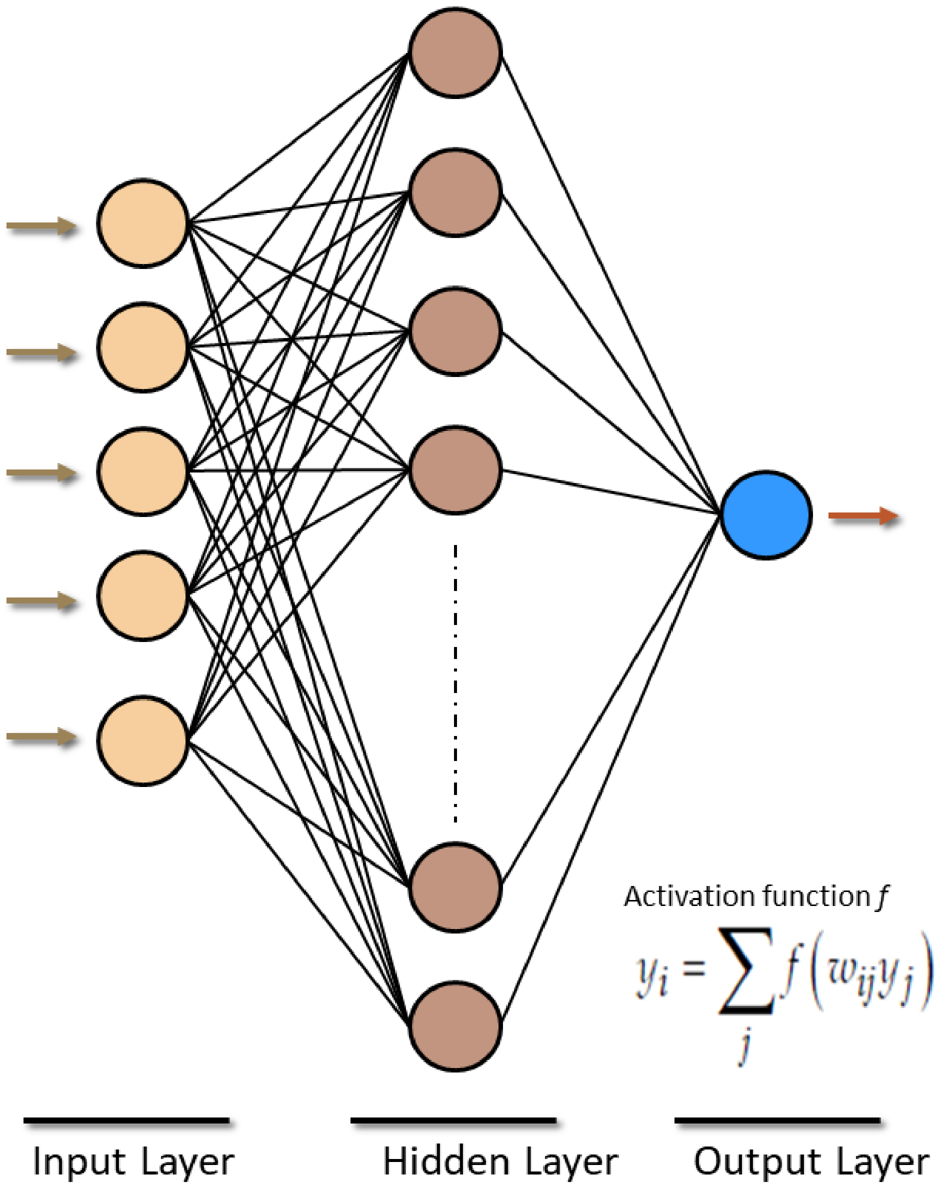


alexa

Neural Networks

Artificial Neural Networks (NN):

- *collection (organized in layers) of interconnected units (nodes)*
- *each node having the capability to receive a signal, process the signal, and transmit the processed signal to other units linked to it.*



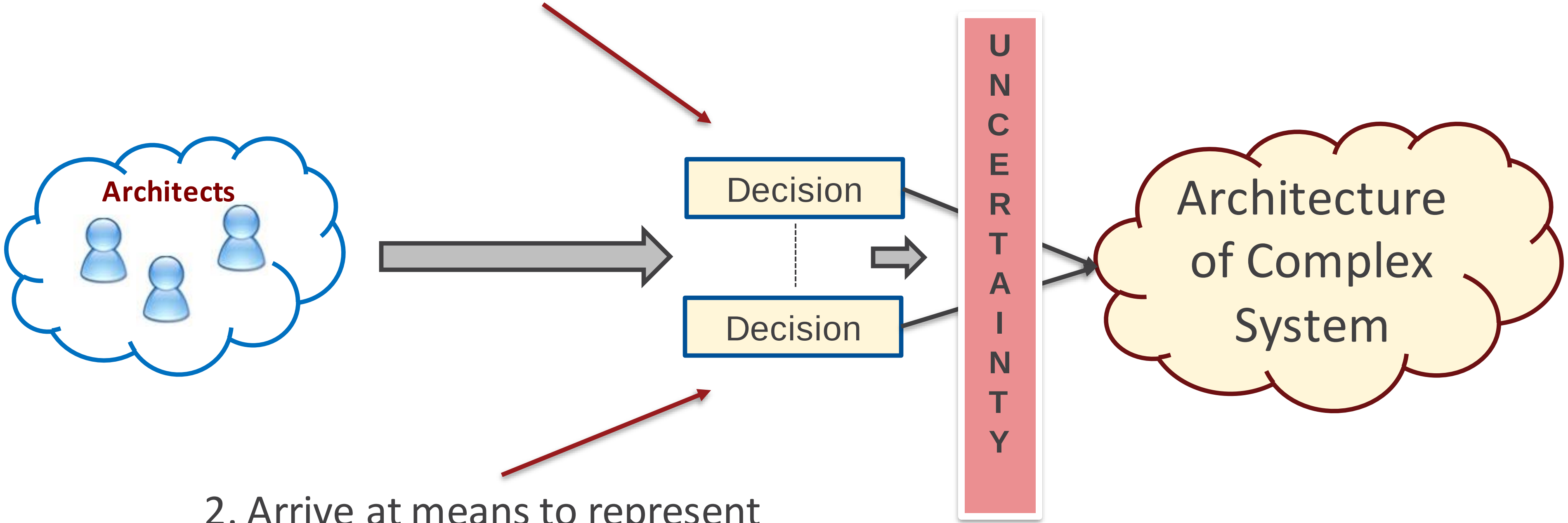
The work presented involves

Dimensionality Reduction

Supervised Learning

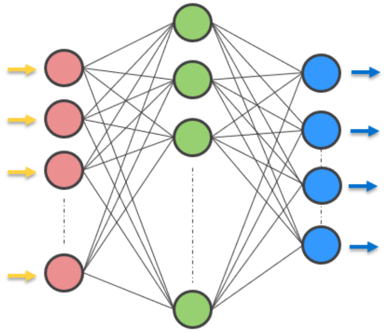
ML approach…

1. Capture the experience architecture design decisions, and label “good” and “not good” decisions



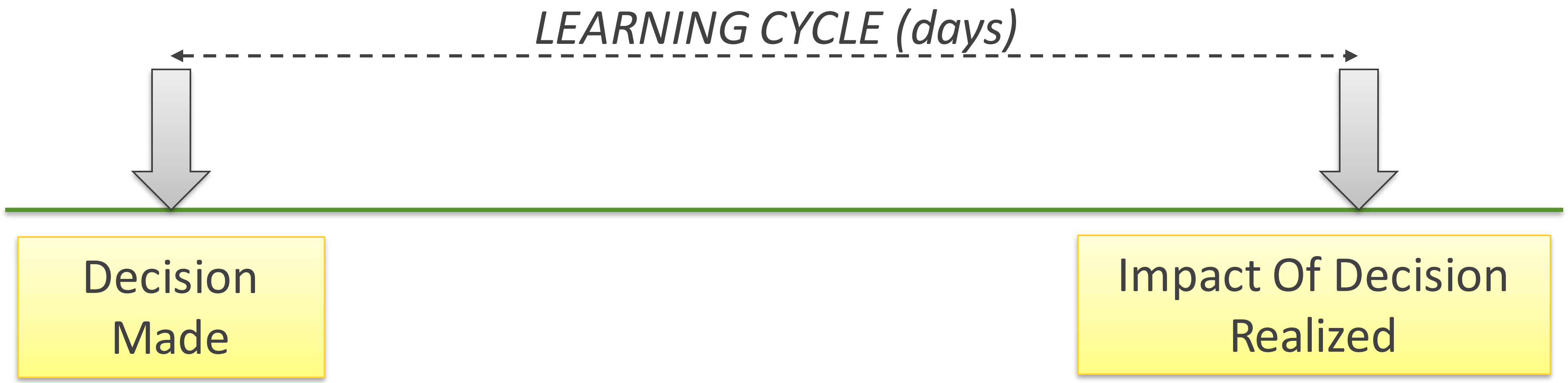
2. Arrive at means to represent decisions – in terms of attributes

3. Train ML Models



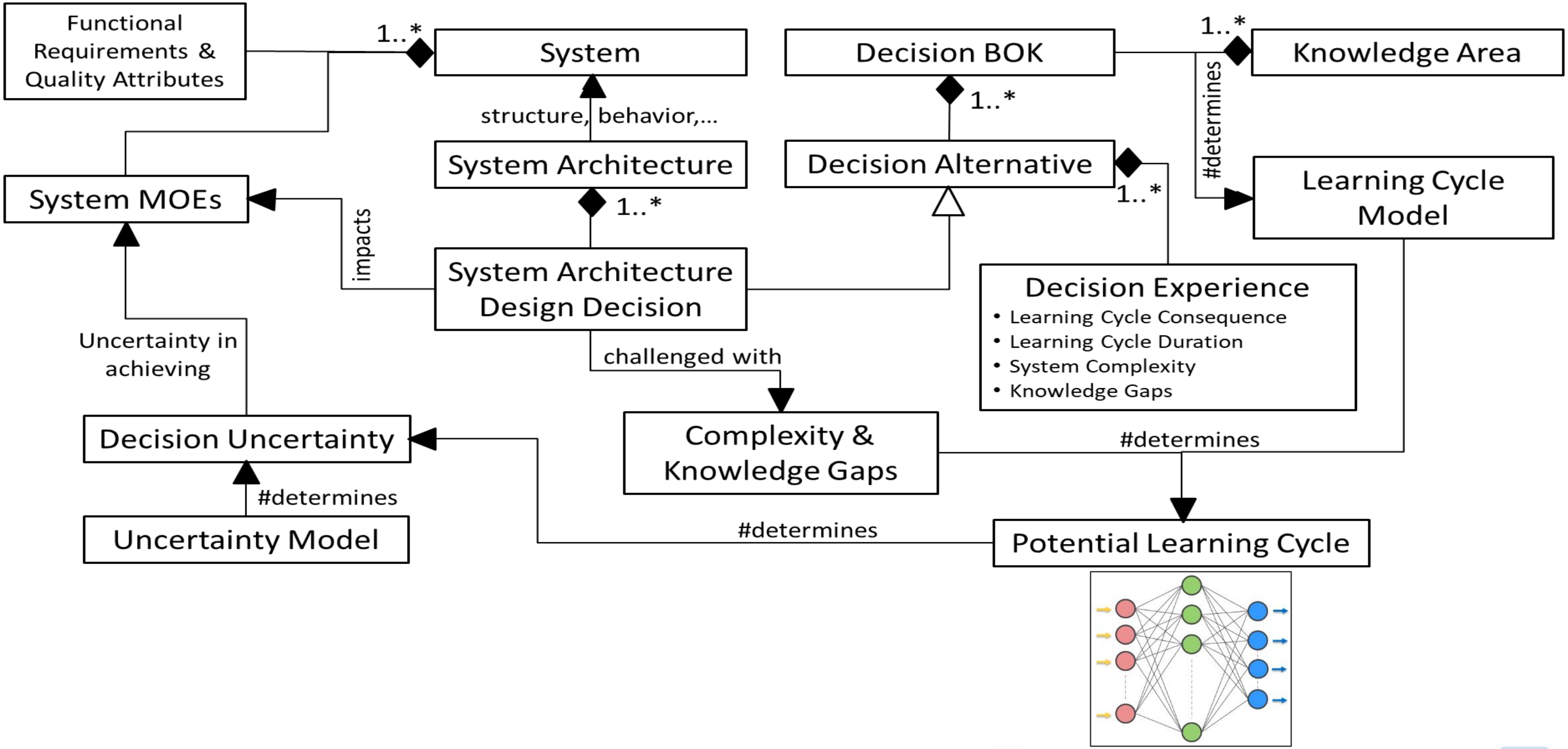
4. Predict & Assess Decision Uncertainty

Learning Cycles



- ❑ Yes, that is the right decision
- ❑ It’s ok, though suboptimal, we can proceed
- ❑ No, we need to patch up (surgery)
- ❑ No, we need to redo/ rework (loop back)

Integrated Model



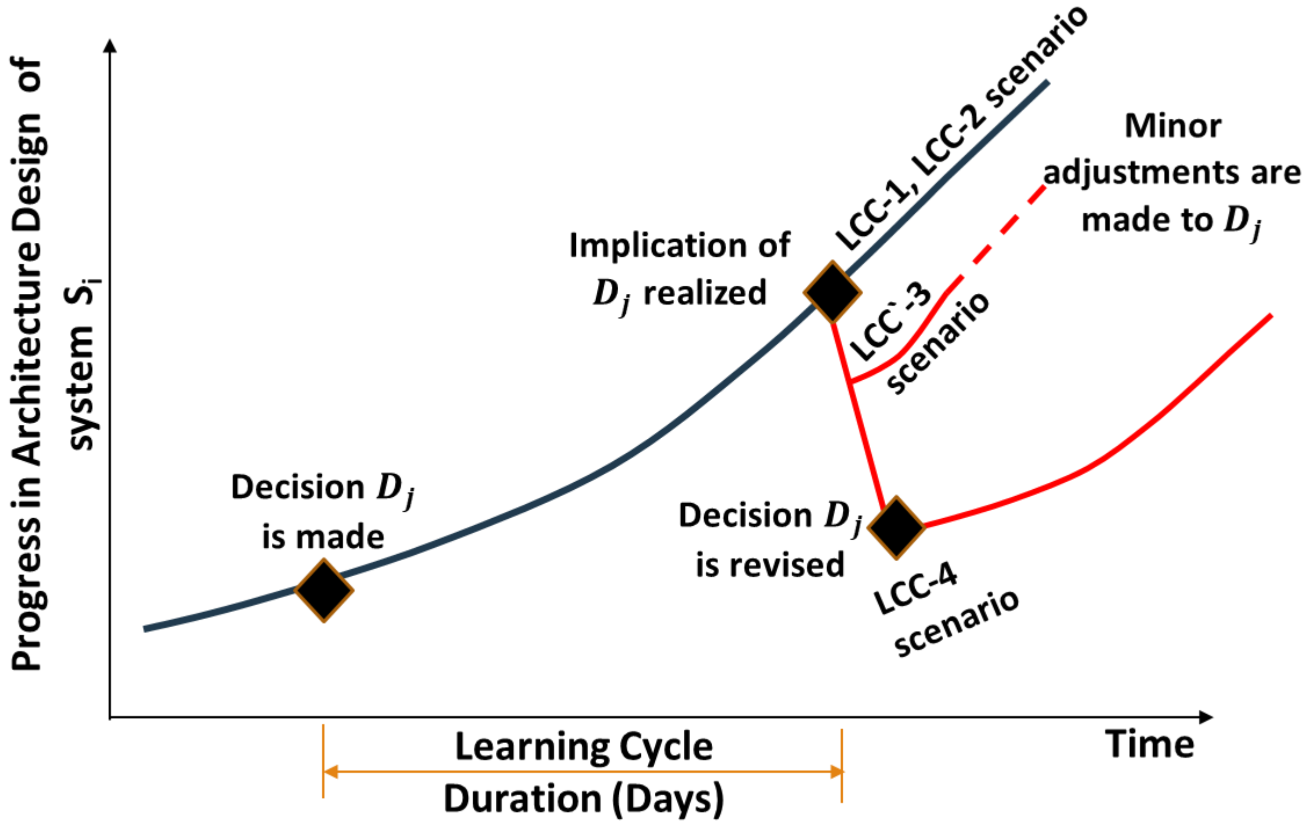
Learning Cycle Consequences & Duration

Learning cycle is indicative of the consequences and duration the design team experiences before realizing the implications of the decisions taken. As design for complex systems and SoS involves decision making with significant uncertainty, the design teams encounter significant number of learning cycles.

Most of the critical architecture design decisions are made in the early development phases, when the prevalent knowledge often is not adequate to make the right decision, the ability to comprehend learning cycles becomes crucial

Learning cycle duration is the duration from when a decision is taken to the point in time when the implications are realized, with a good degree of certainty. Learning cycle consequence indicates the type of consequence encountered for the decision taken

Learning Cycle Consequences



LCC-1	The decision is the optimal decision, and does not inhibit any of the requirements or expected behaviors of the system
LCC-2	The decision is not the optimal one, but nevertheless, can be "lived with", i.e. does not impact any critical requirements or behaviors of the system
LCC-3	The decision is not optimal, and it might require some amount of rework, minor correction or "surgery"
LCC-4	The decision needs to be significantly reworked, requiring a loop-back to the point where the decision was taken. In extreme cases, the budget or resources required for the rework might be far in excess of what is available or allowable for development

Learning Cycle Consequence

Two Phases – Learning & Prediction

Feedback the architecture design decision experiences

Learning Phase

- Codify architecture design decision experiences
- Analyze Learning Cycle Space
- Train the Learning Cycle Model [machine learning]
- Formulate Uncertainty Model

Prediction Phase

- For new architecture design decision, codify the alternatives considered
- Use Learning Cycle Model to predict potential learning cycles
- Use Uncertainty Model to assess uncertainty and factor in decision for choosing a specific alternative
- Monitor & reassess decision uncertainty with time

Provide Learning Cycle Model & Uncertainty Model

Learning Phase

L1: Decision Data Collection

a) Analyze decisions taken for prior system arch design
b) Represent alternatives chosen in terms of decision attribute values
c) Incorporate knowledge gaps when decision was taken
d) Represent system complexity of the prior system
e) Capture corresponding learning cycle for the decisions

L2: Analyze Learning Cycle Space

a) Dimensionality reduction & clustering of decision alternatives
b) Dimensionality reduction of complexity & knowledge gaps
c) Visual model of Decisions Alternatives vs Decision Complexity vs Learning Cycle (Consequences, Duration)

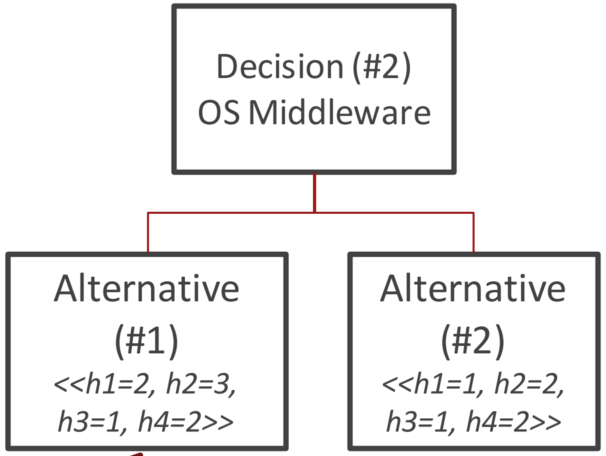
L3: Train Learning Cycle Model

a) Formulate Machine Learning model with decision attributes, system complexity and knowledge gaps as features, and learning cycle as targets
b) Split data set (feature target) into training, cross validation and test sets
c) Train model, evaluating Prediction Accuracy, and establish baseline model

L4: Formulate Uncertainty Model

a) Analyze required uncertainty levels based on learning cycle consequences & duration
b) Formulate uncertainty model surface based on relevant organization & team factors

Codification of Decision Experience



Alternative #1 was used in a system with complexity measure 4. There was 8 knowledge gaps that existed when the decision was made. The decision was found to be optimal (LCC-1) and this was realized after a long duration

<u>Decision #2: OS Middleware : Alternatives - Attributes</u>

h1 Scheduling: co-operative = 1, rate monotonic=2,…

h2: NVM writing: periodic = 2. on demand = 3,…

h3: Buffer: fixed = 1, variable=2, ring buffer=3,…

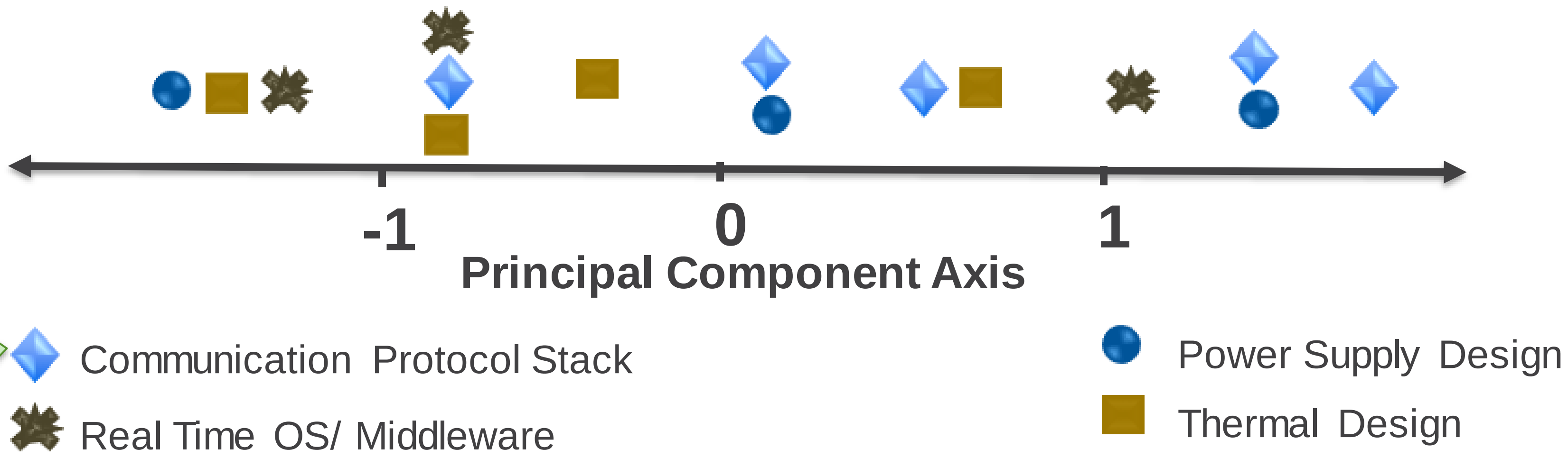
h4: thread life management: ….

Codification of Learning Cycle Experiences pertaining to Architecture Design Decisions

Decision ID	h1	h2	h3	h4	Sys Comp	K-Gaps	LCC	LCD
2	2	3	1	2	4	8	1	3
1	1	2	2	1	3	12	3	2
3	2	3	1	2	1	5	3	2
1	1	2	2	1	2	6	2	1
2	1	2	1	2	3	11	4	3

PCA of Decision Alternatives

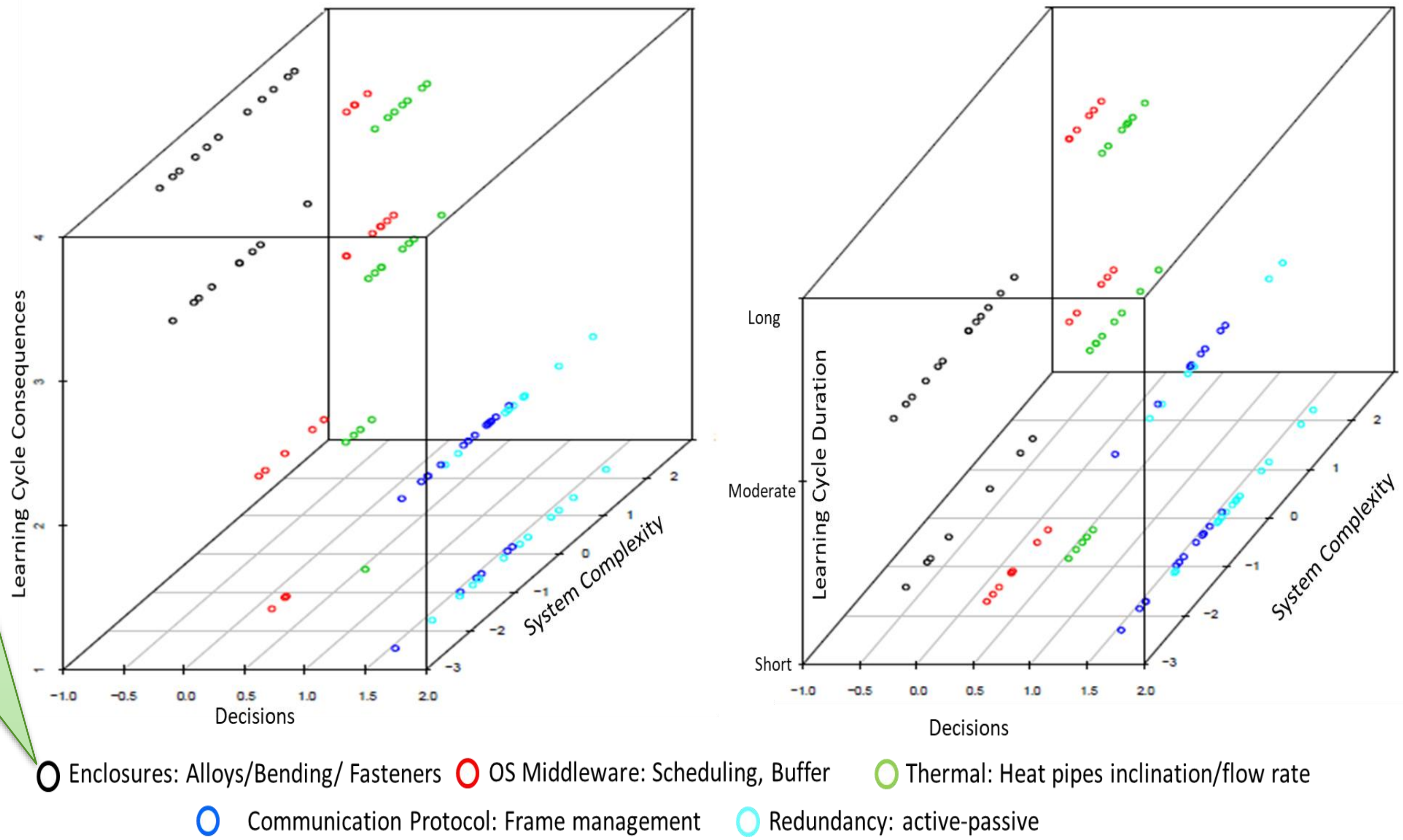
All the decisions taken pertaining to Communication Protocol Stack manifest as five distinct clusters (-0.81, 0.14, 0.54, 1.57, 1.82 on the Principal Component Axis),



- PCA adopted to reduce the dimensionality of the multi-dimensional decision space
- Clustering of the decision points along the reduced single dimension of the decision space enables easy visualization and analysis in terms of distinct clusters and characteristics

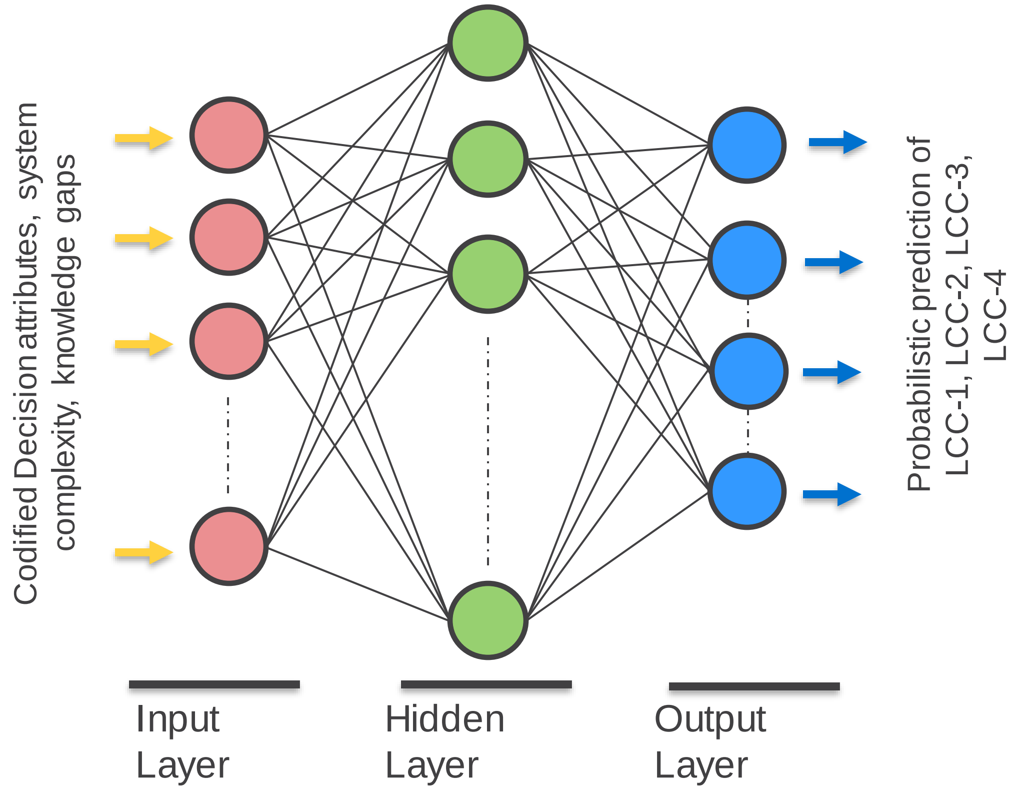
Learning Cycle Space

Decisions pertaining to Enclosures: Alloys/ Bending have mostly experienced LCC-3 and LCC-4, across a wide range of varied system complexity and knowledge gaps (low and high).



- Provides insights into the architecture design decision-making experience
- Represents the experiential knowledge pertaining to the various architecture design decisions taken for different systems

ML Model Training



Activation Function

$$sigmoid(z) = g(z) = \frac{1}{1 + e^{-z}}$$

$$g'(z) = \frac{d}{dz} g(z) = g(z)(1\text{-}g(z))$$

Input Layer: $a^{(1)} = x$

Hidden Layer: $z^{(2)} = \Theta^{(1)} a^{(1)}$

$a^{(2)} = g(z^{(2)})$

Output Layer: $z^{(3)} = \Theta^{(2)} a^{(2)}$

$a^{(3)} = g(z^{(3)}) = h_\theta(x)$

- Machine Learning methods adopted to train the Learning Cycle Model
- The past decision experiences in the Learning Cycle Space form the Training Set
- The Learning Cycle Model learns about the learning cycles experienced, pertaining to various decisions taken over the period of development and evolution of various systems in the organization

ML Model Training

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{4}[-y_k^{(i)} log((h_\theta(x^{(i)}))_k - (1-y^{(i)}) log((1-h_\theta(x^{(i)}))_k] + \frac{\lambda}{2m}[\sum_{j=1}^{25}\sum_{k=1}^{9}(\Theta_{j,k}^{(1)})^2 + \sum_{j=1}^{4}\sum_{k=1}^{25}(\Theta_{j,k}^{(2)})^2]$$

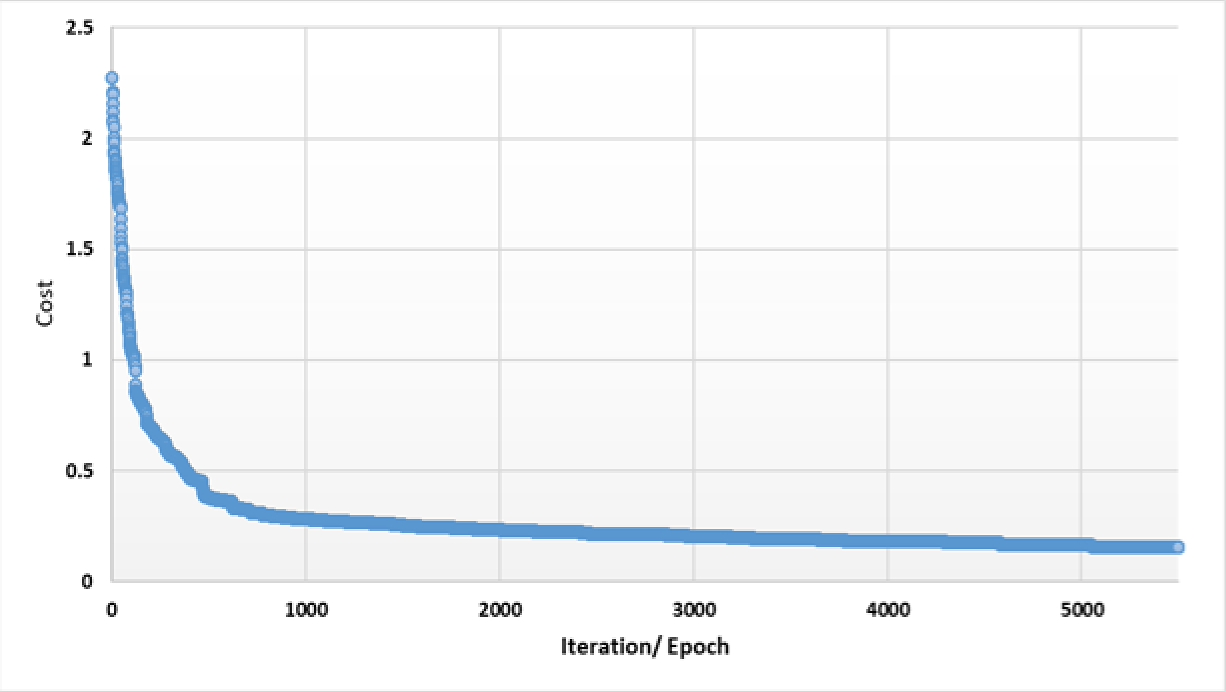
The neural network used had one hidden layer comprising 25 neurons with 9 input layer units

- Learning algorithms were devised that can automatically tune (and learn) the weights and biases associated with various neurons, so that the output produced by the network closely matches the desired classification of the learning cycle consequences and duration
- Mathematically, this close matching involves an associated cost function that needs to be minimized. Hence, the training process is iterative, to minimize the cost function below a threshold, with each iteration fine tuning the parameters

ML Model Prediction Performance

PREDICTION	LCC-1	LCC-2	LCC-3	LCC-4	
LCC-1	27.4%	0.8%	0.0%	0.0%	97.3% 2.7%
LCC-2	0.4%	30.5%	0.0%	0.0%	98.8% 1.2%
LCC-3	0%	0.8%	15.8%	0.4%	93.2% 6.8%
LCC-4	0%	0.4%	0.4%	23.2%	96.8% 3.2%
	98.6% 1.4%	94.0% 6.0%	97.6% 2.4%	98.4% 1.6%	96.9% 3.1%

TARGET



ML Model Prediction Performance

- The rows correspond to the predicted learning cycle consequences, while the columns correspond to the actual learning cycle consequences as part of the training set (Target).
- The diagonal cells indicate the correctly classified observations. The off-diagonal cells are the incorrectly classified observations. Each cell in (4×4) colored matrix indicates the percentage of the total observations.
- The rightmost column indicates the percentages of all the training set predicted to belong to each class that are correctly and incorrectly classified. These metrics are the precision (or positive predictive value) and false discovery rate, respectively.
- The bottom-most row shows the percentages of all the training set belonging to each class that are correctly and incorrectly classified. These metrics are called the recall (or true positive rate) and false negative rate, respectively.
- The cell in the bottom right of the plot shows the overall accuracy. For instance, there were 23.6% records in the training set that had LCC-4 as the consequence experienced. The machine learning model performance was that 23.2% were predicted correctly to be LCC-4

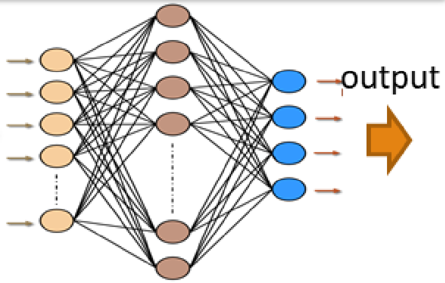
PREDICTION					
LCC-1	27.4%	0.8%	0.0%	0.0%	97.3% 2.7%
LCC-2	0.4%	30.5%	0.0%	0.0%	98.8% 1.2%
LCC-3	0%	0.8%	15.8%	0.4%	93.2% 6.8%
LCC-4	0%	0.4%	0.4%	23.2%	96.8% 3.2%
	98.6% 1.4%	94.0% 6.0%	97.6% 2.4%	98.4% 1.6%	96.9% 3.1%
	LCC-1	LCC-2	LCC-3	LCC-4	
		TARGET			

Validation of Learning Cycle Model

Decision Experience pertaining to architecture design decisions

DecisionID	h1	h2	h3	h4	SysComp	K-Gaps	LCC
2	2	1	1	2	3	11	4
1	3	2	1	1	1	5	1
3	3	1	2	2	3	6	4
1	1	2	2	1	3	9	3
2	2	3	1	2	4	9	2

input



Learning Cycle Model

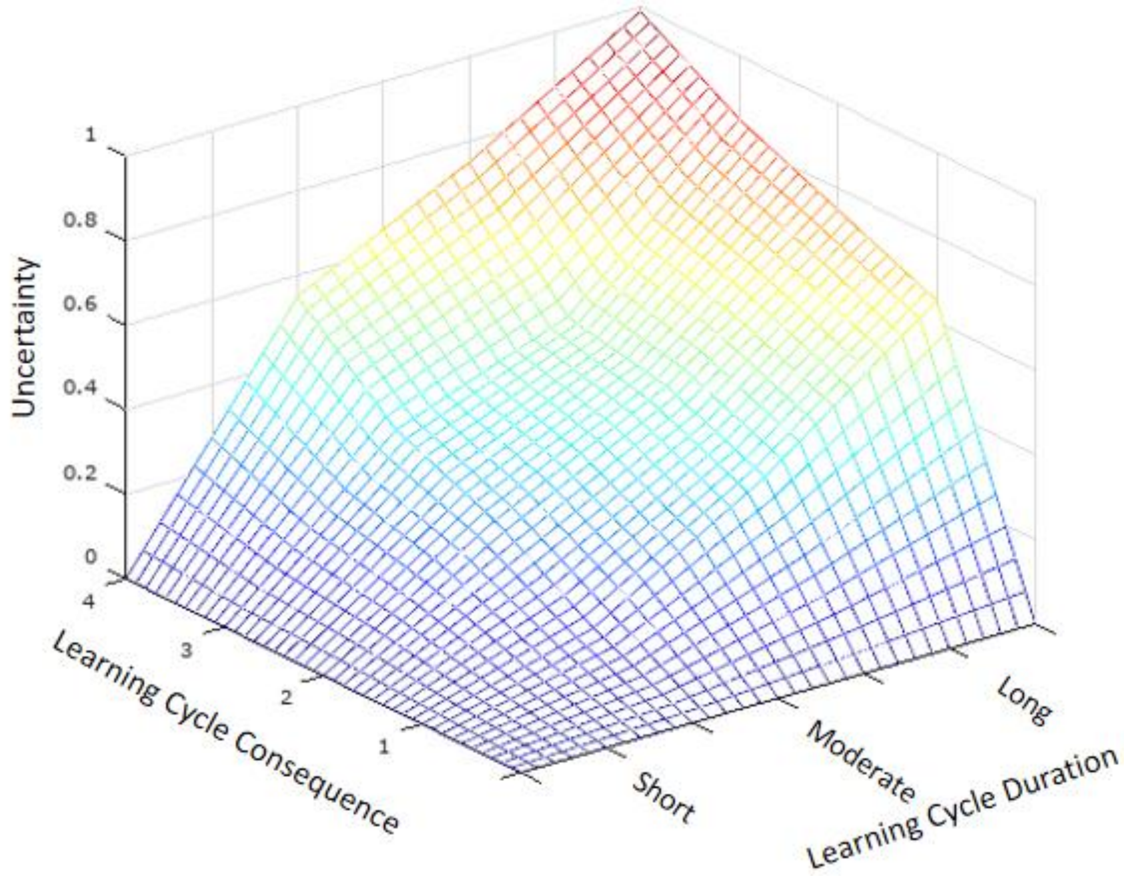
Probabilistic predictions of Learning Cycle Consequences

LCC-1	LCC-2	LCC-3	LCC-4
0.0016817217	0.0061270562	0.0000796866	0.9549523432
0.9999969555	2.3560401E-11	2.2714304E-06	0.0000422213
0.0000170234	1.2163202E-11	1.2490132E-10	0.9997610606
0.0004465759	0.0008018442	0.9963371391	0.0072283499
0.0342013134	0.9708418369	6.1549942E-07	0.0001666969

- Learning Cycle Consequence prediction: Validation is in terms of the highest probability prediction for the specific learning cycle consequence, in tandem with the actual experience

- Similar approach is done to predict learning cycle duration, categorized as short-moderate-long

Uncertainty Model



- Built as a surface that is formulated based on the Learning Cycle Consequences and Learning Cycle Duration
- Higher uncertainty (maxima point on the uncertainty surface) is associated for LCC- 4 with long Learning Cycle Duration
- The lowest uncertainty (minima point on the uncertainty surface) is associated with LCC-1 with short Learning Cycle Duration
- A team or organization to appropriately calibrate the uncertainty surface based on factors such as knowledge areas, culture, performance of the teams and organizational stage gate processes

Prediction Phase

P1: Architecture Design Decision Alternatives

a) For architecture design decisions, arrive at the possible alternatives

b) Represent alternatives considered in terms of decision attribute values. Identify knowledge gaps for decision, and system complexity

c) Codify the alternatives, system complexity and knowledge gaps

P2: Predict Potential Learning Cycle

a) Input the codified alternatives, system complexity and knowledge gaps to Learning Cycle Model

b) Learning Cycle Model predicts the potential learning cycles for the considered alternatives

P3: Uncertainty Assessments & Visualization

a) Use Uncertainty Model to asses the uncertainty of decision

b) Analyze uncertainty assessments and factor into deciding the specific alternative

P4: Refresh Uncertainty Assessments

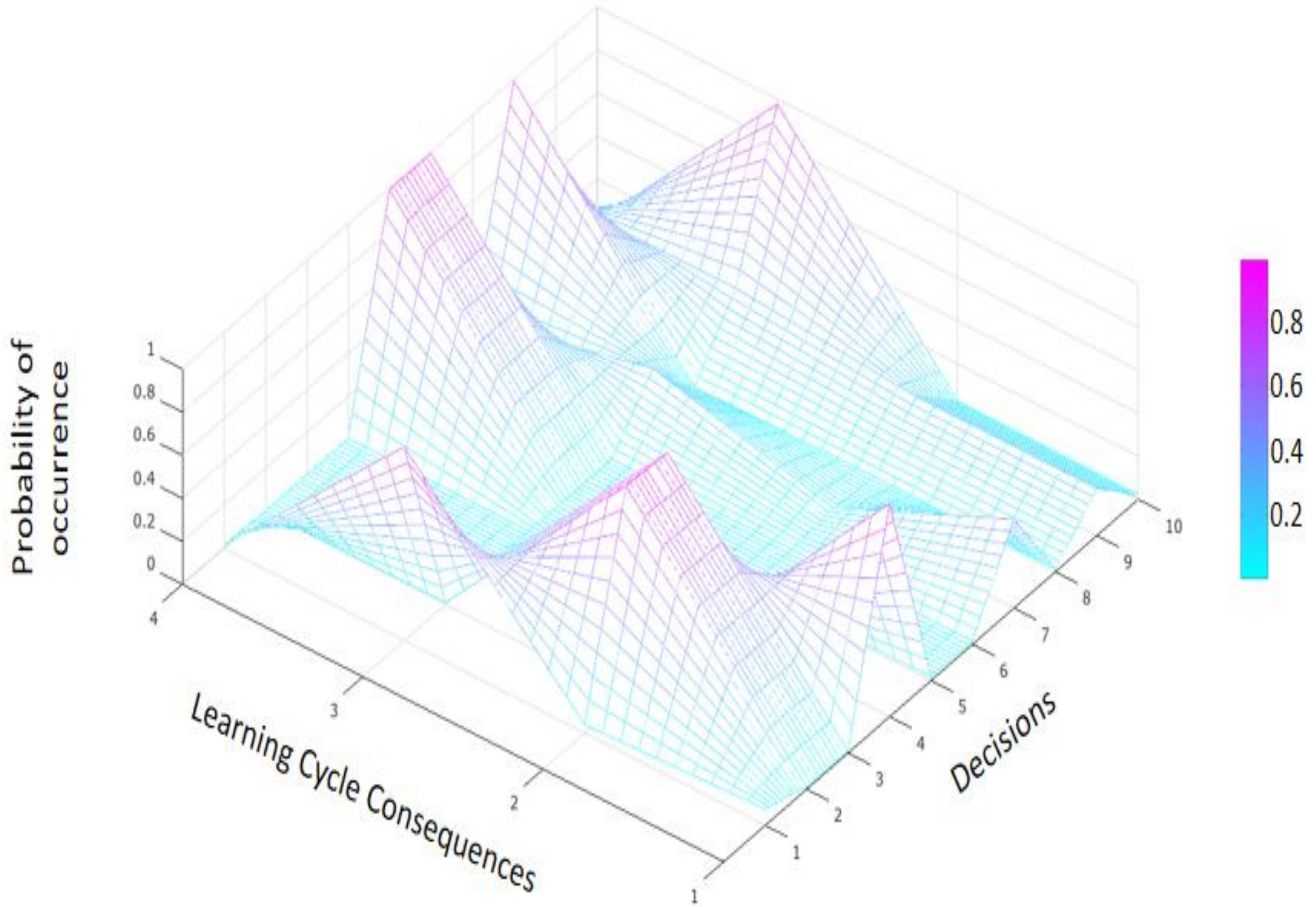
a) As lifecycle progresses, revise knowledge gaps, to get revised inputs

b) Get revised potential learning cycle P2 and uncertainty assessment P3

P5: Feedback to Learning Cycle Model

a) When learning cycle is encountered for the decisions, include the learning cycle experiences in Decision Data (L1) and refresh L2, and retrain L3

LCC Prediction through ML Model

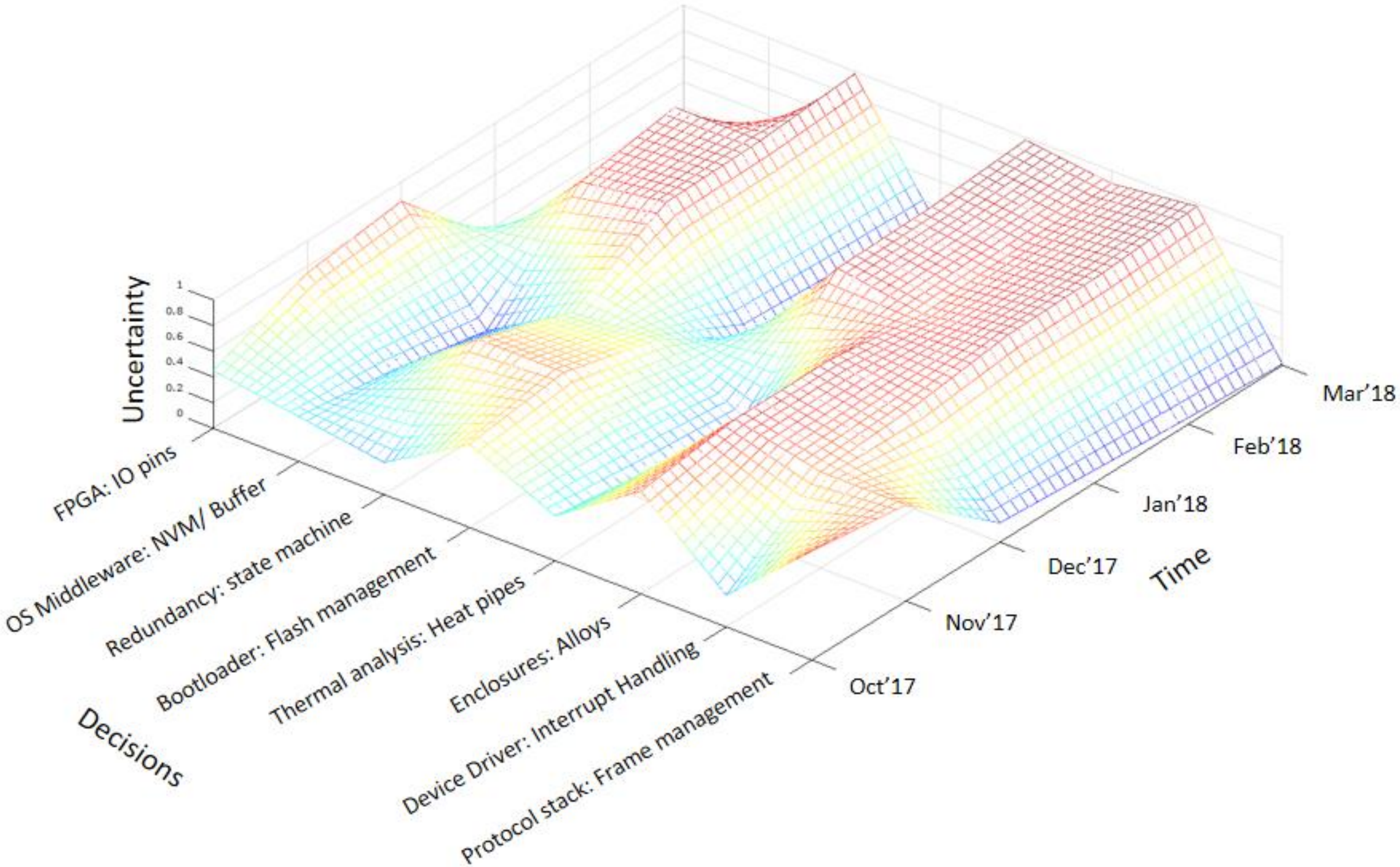


- For architecting the system, various decisions and corresponding feasible alternatives are enlisted, along with the corresponding knowledge gaps

- The decision-making process requires the architects to analyze the set of possible alternatives pertaining to each decision

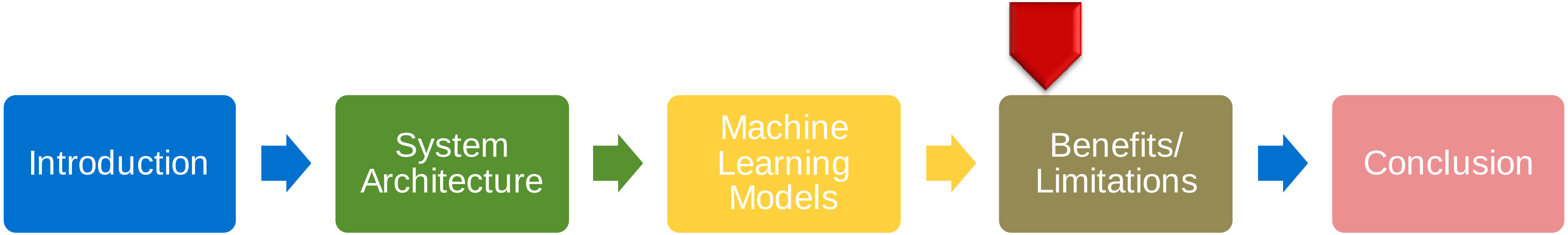
- The Learning Cycle Model trained in [L2] is used to predict the potential learning cycles for the shortlisted alternatives

Decision Uncertainty Assessment & Monitoring



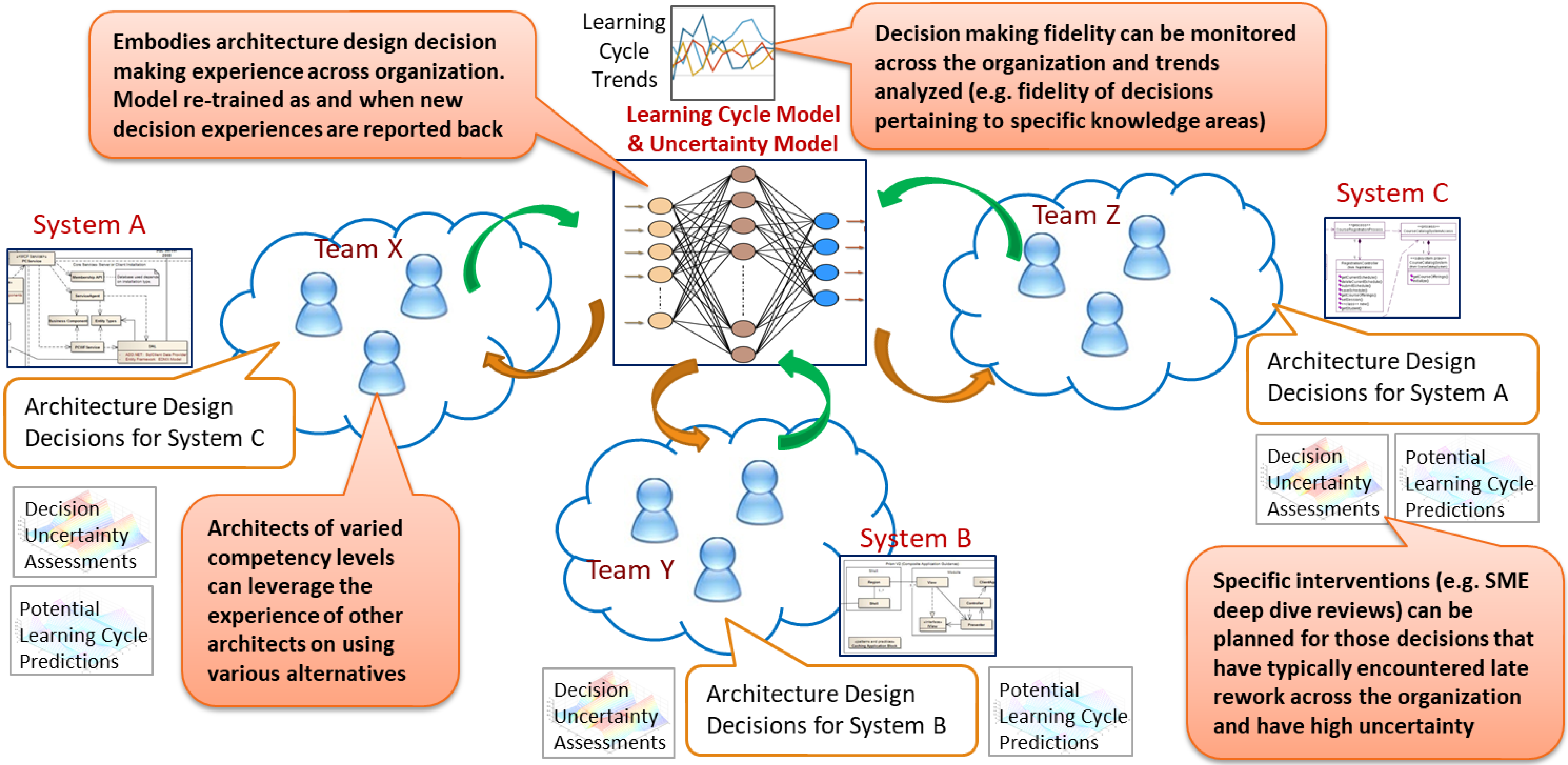
- ❑ Based on the learning cycle consequence and duration probabilities for the selected alternative, the Uncertainty Model formulated in [L3] is used for assessment of the corresponding uncertainty
- ❑ As the development progresses, the uncertainty is to be re-assessed since there will be changes in the knowledge gaps associated with the decision

System Architecture



- Benefits
- Limitations

Benefits



Embodies architecture design decision making experience across organization. Model re-trained as and when new decision experiences are reported back
Learning Cycle Trends
Learning Cycle Model & Uncertainty Model
Decision making fidelity can be monitored across the organization and trends analyzed (e.g. fidelity of decisions pertaining to specific knowledge areas)
System A
Team X
Team Z
System C
Architecture Design Decisions for System C
Architecture Design Decisions for System A
Decision Uncertainty Assessments
Potential Learning Cycle Predictions
Architects of varied competency levels can leverage the experience of other architects on using various alternatives
Team Y
System B
Decision Uncertainty Assessments
Potential Learning Cycle Predictions
Architecture Design Decisions for System B
Decision Uncertainty Assessments
Potential Learning Cycle Predictions
Specific interventions (e.g. SME deep dive reviews) can be planned for those decisions that have typically encountered late rework across the organization and have high uncertainty

Limitations

- The architecture design decision making experience is required to build the Learning Cycle Space and train the Learning Cycle Model
  - In case these data points are not available explicitly, part of it can be elicited through interviews and group discussions with architects
  - Else, mechanisms for collecting and codifying the upcoming decision experience can be setup, so that over a period, these data points are available
- As new architecture design decision experience arrives, the Learning Cycle Space needs to be reconstructed, and the Learning Cycle Model needs to be retrained every time
  - This can however be automatically triggered, to facilitate availability of the experience at the earliest to architects across the organization

Limitations

- Each decision is represented through a set of attributes, and the alternatives corresponding to the decision would have different values for those attributes.

- Codification of the attributes and the associated values might need to be fine-tuned over time, to take into consideration evolution of the knowledge of the alternative
  - For instance, initially it might turn out that three attributes are adequate to represent a decision. Later, it might turn out an additional attribute is required, due to discovery of a new alternative for the decision

Conclusions

- *Systems approaches and systems thinking are critical for the design & development of large, complex, & trustworthy systems*
- *This Tutorial described an approach to leverage Machine Learning to learn from architecture/ design decision learning cycles & assess uncertainty of decisions*
- *The approach enables progressive maturity of the architectural knowledge base and aid robustness in architecture design decisions*

AUGMENT THE INTELLIGENCE OF SYSTEM ARCHITECTS & DESIGNERS

For more details…

SYSTEMS ENGINEERING
The Journal of The International Council on Systems Engineering
INCOSE

REGULAR PAPER

Decision learning framework for architecture design decisions of complex systems and system-of-systems

Ramakrishnan Raman, Meenakshi D'Souza

First published: 07 November 2019 | https://doi.org/10.1002/sys.21517 |

Volume 22, Issue 6
SOSE Special Issue
November 2019
Pages 538-560

THANK YOU