Paper Presentation

Dynamic Evolution of Context and Corresponding Self Configuration in Service Oriented Cyber Physical System

Presented by : Muhammad Salman Aslam (G-00763290)

Department of Computer Science George Mason University

(12/04/17)

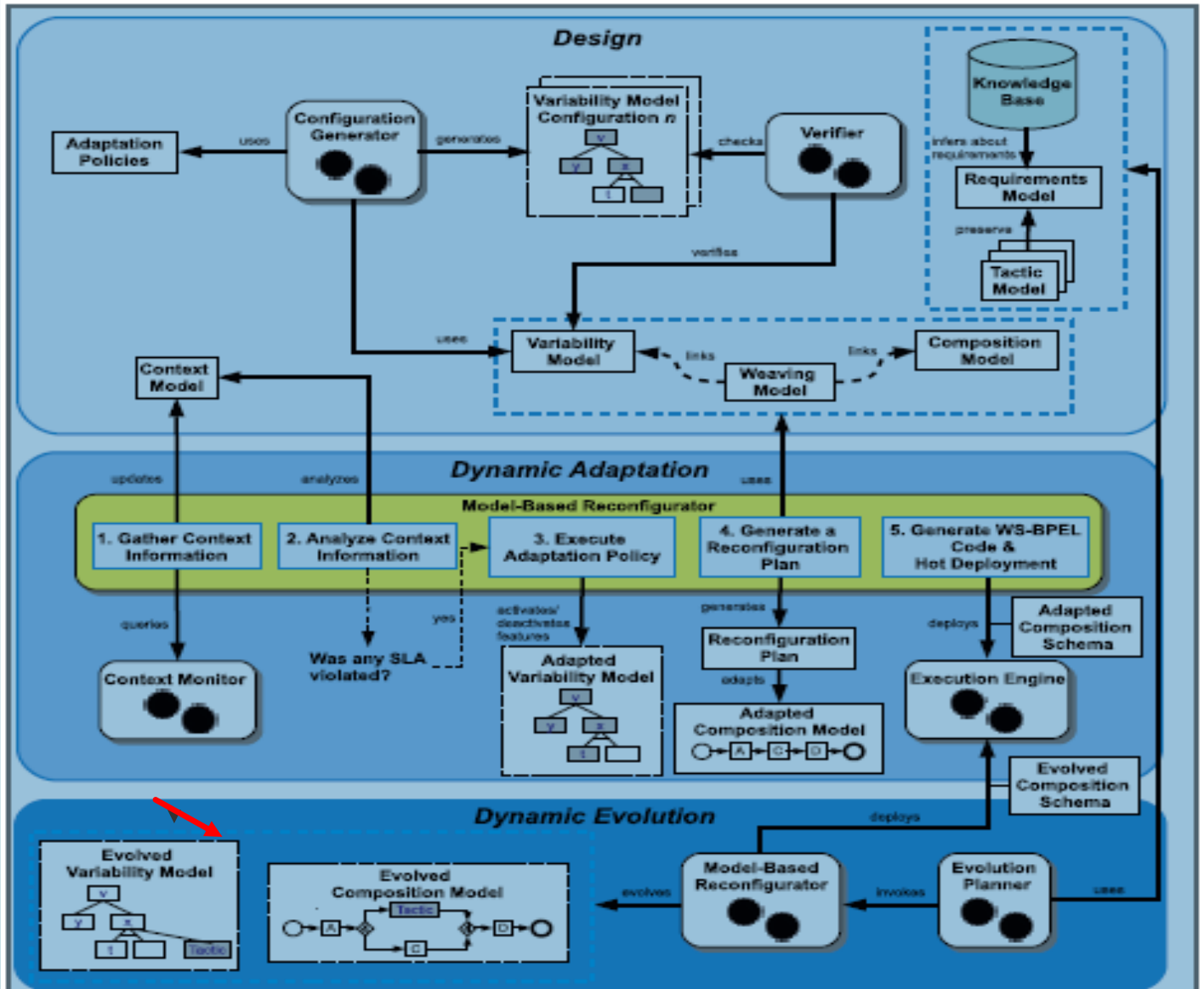
SOA and Optimal Service Composition

- Composed of services
  - Distributed Design
  - Standard interfaces and communication protocols
  - Loosely coupled services
  - Functional Demarcation

- Optimal Selection of SOA architecture
  - Selection if service components
  - Selection Service providers
  - Re configuration and re architecting
  - Optimal Allocation to meet SLAs
  - Self Configuration - adaptation with changing context

SOA Adaptation Model.

- Layered SOA Optimization model described by Alferiz et.al. [5]
  - Design Phase
    - Requirements/Traffic Model
    - Variability Model
    - Configs gen
    - Context
    - Weaving
  - Adaptation Phase
    - Adapted Variability
    - Adapted Composition
  - Evolution phase
    - Evolves Composition
    - Evolve Variability



Context, CAS

- "the circumstances that form the setting for an event". Alagar et.al.
- Any information that represents a change in an entity important to a systems
  - Change in Environment
  - Change in User
  - Change in computing platform
- Context Aware Systems
  - Sense and compose context information from different sensors;
  - Autonomously detect relevant changes in the context in order to dynamically adapt their services;
  - Interoperate with third-party service providers

Where is the Context Most Important

- Context is most important for systems where
  - Dynamic
  - Distributed
  - Users may be added. User roles may Change
  - Actions and activities can not be all known priori
  - Decision making is non deterministic – needs Intelligence
  - Large systems with many interconnection
  - Distributed applications and services
  - Environment changes are large and unpredictable
    - High Interaction with the Physical world

- Characteristics of Cyber Physical Systems

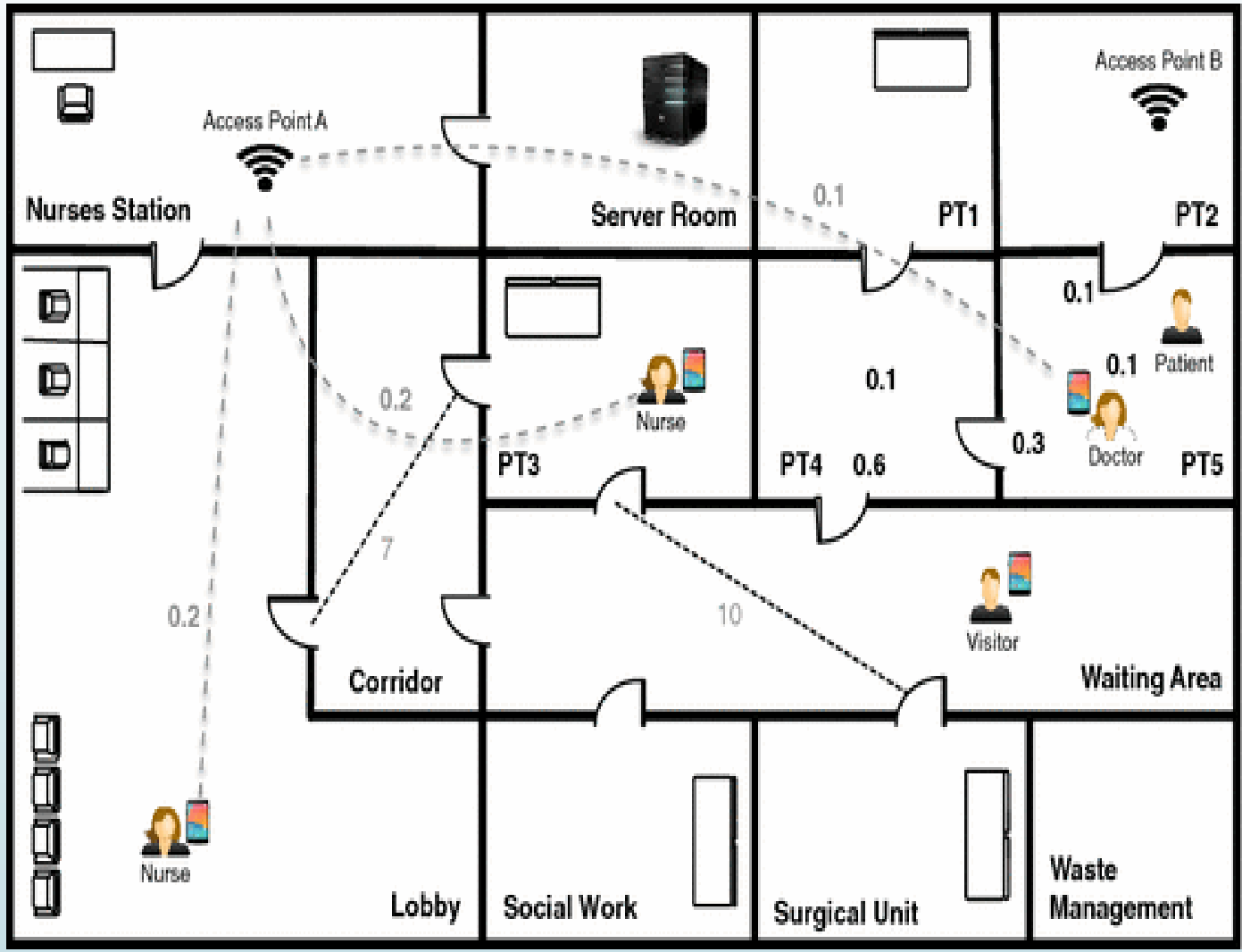
Characteristics of CPS

- Interact with environment
- Heavily Networked
  - Components interconnected
- Distributed
  - Components spatially spread
- Integrated
  - Tightly coupled components (usually)
- Time Constrained (Real Time)
  - Hard or Soft Real time constraints
- Dynamic / Configurable
- Dependable/Reliable
- Secure

Smart Hospital Spaces

Importance of Context in CPS

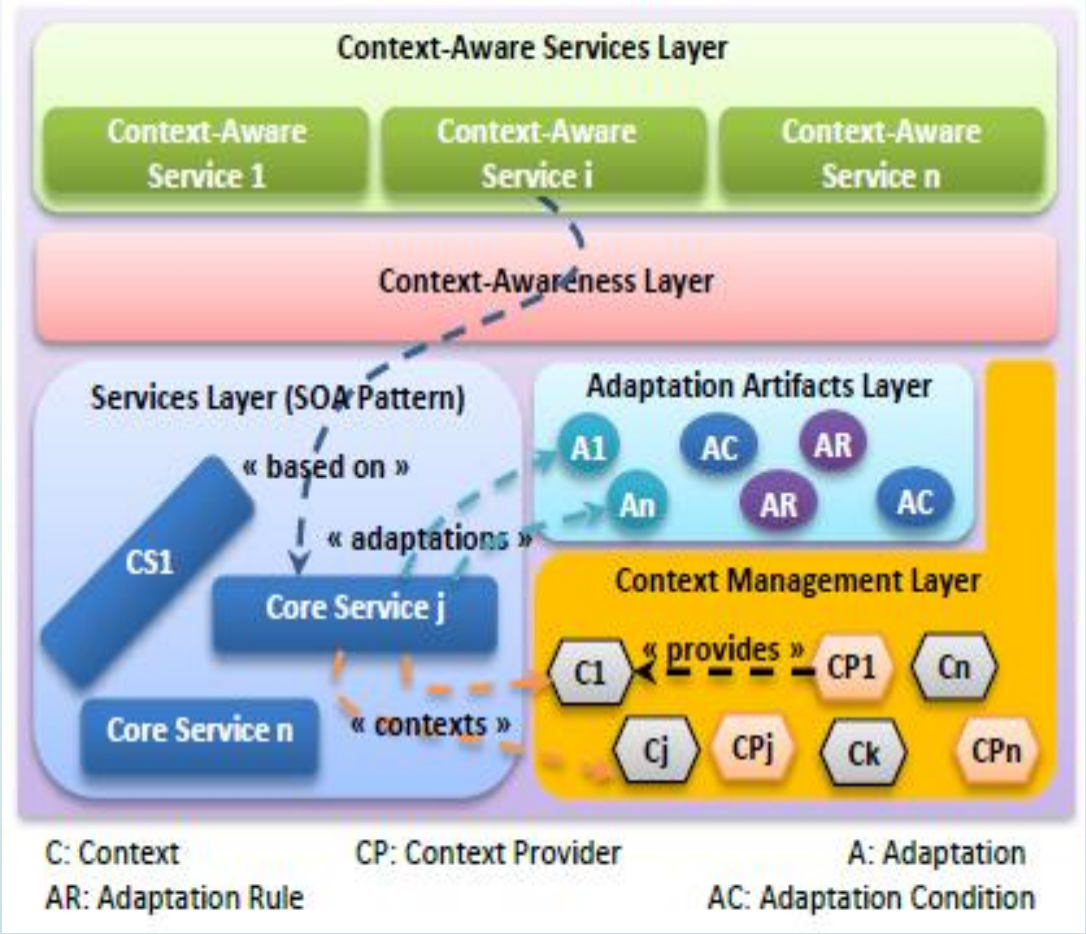
- Consider Reconfiguration Services
  - Patients Population has gown the ward needs expansion
  - Before moving the wall consider whether the equipment is all off and ready to move + no occupancy in adjacent rooms + Authorization etc
- Navigation Service
  - Service routing visitors and staff
  - Consider Critical patient in the Hallway
  - In this Context Nav Service needs to advise users to stop
  - What if there is an Emergency Doctor using he Nav application?



Model to Derive Context and Evolve Context Knowledgebase

Where to model the Context

- Context Aware Service compositions
- Service Providers
- Service
- Middle ware
- Application
  - Coordinator
  - Eg. Smart car, Smart space



Using SOA Adaptation for Context Evolution

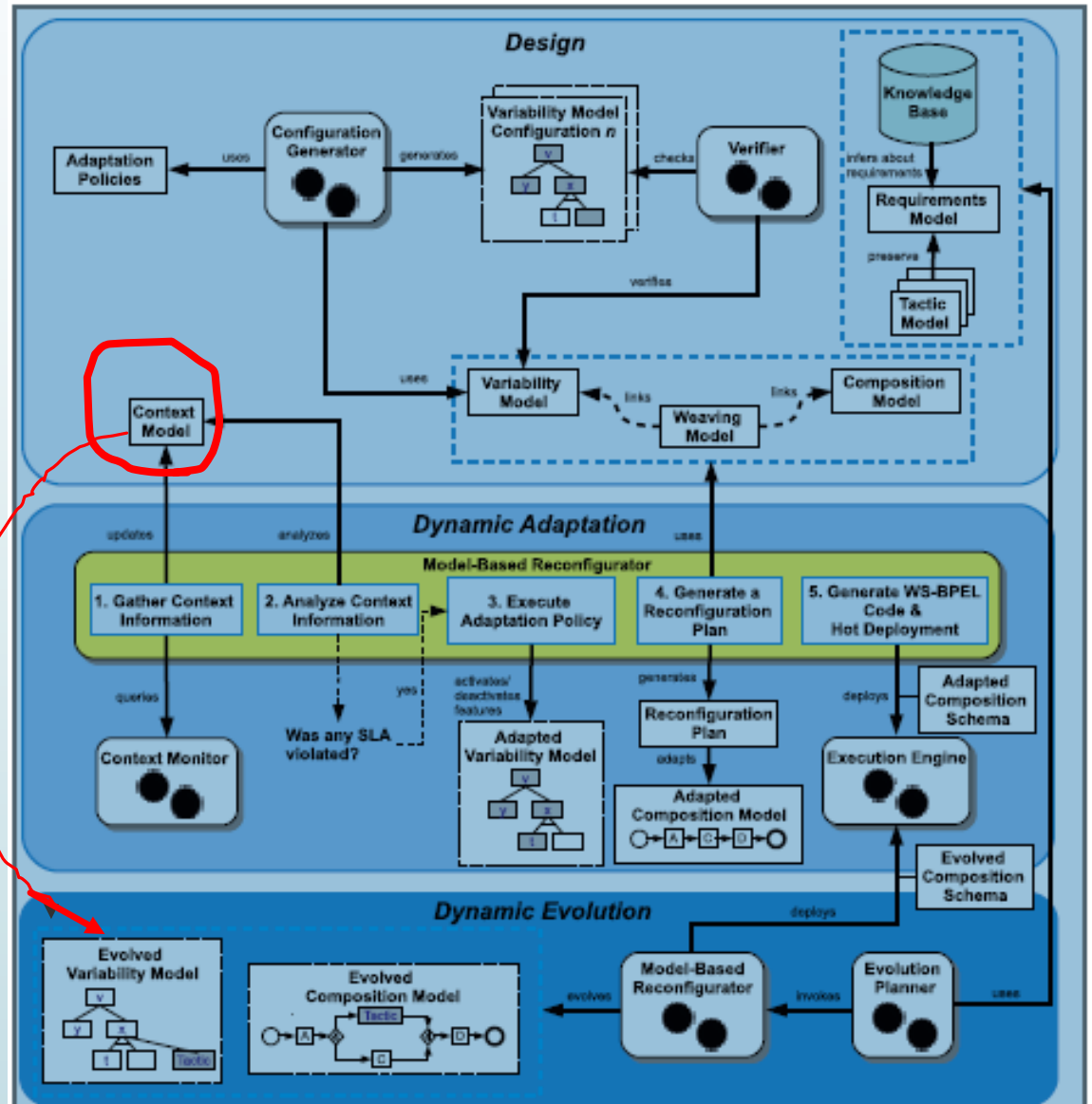
- The figure shows he Fragmented SOA Adaptation Model.
- Context model is built in a closed world Assumption (Circled Red)
- We need to move the Context Model to open world assumption and move to the Dynamic Evolutionary layer.
- How to Evolve Context ?

Context = Known Context $C_{(k)}$ U Unknown Context

Context $C_{(k)}$ = ∑unknown context events that can be handled

Unknown Context = ∑unhandled unknown context events

- $C_{(k)} = C_{(k)} \; U \; c_{(new)}$



How to model the Context

- **Various Ontologies – we Used one by Hatim et. Al [6.10]**
  - Model using simple constructs

    Number of occupants >0

    Light – Low /Off

    Machine X - Ready

Various levels - context divided in to categories

User

Environment

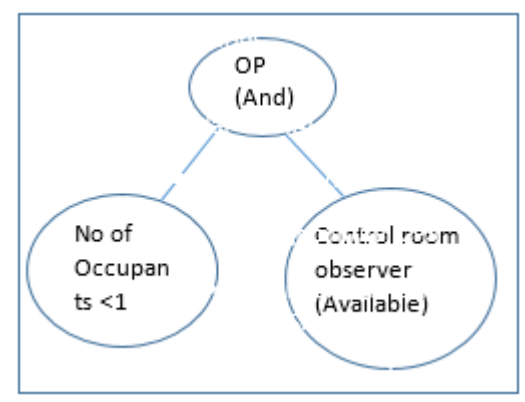
Devices

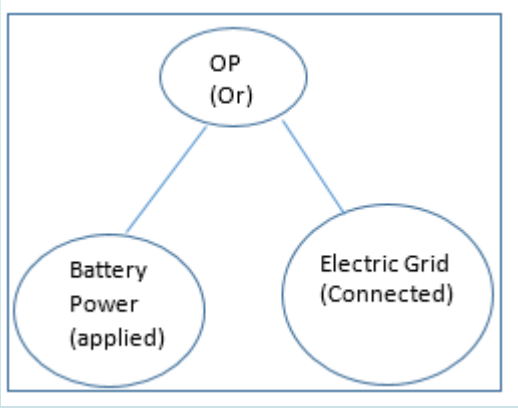
Application

S

How to Model Context

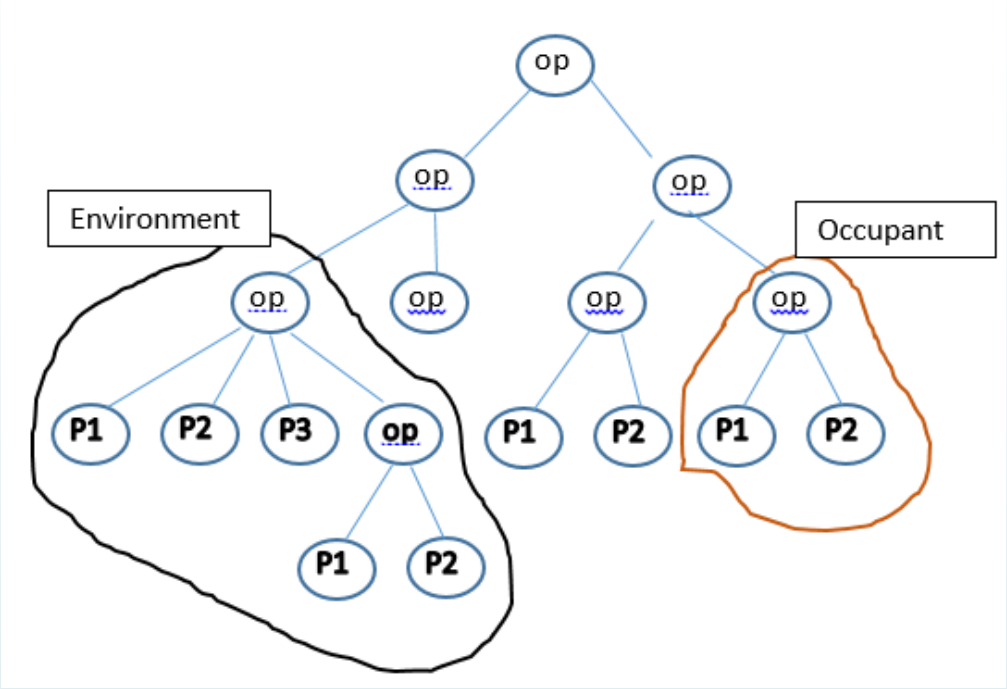
- **Captured In simples Propositions**
  - Context Properties
  - Joined together by Operator
- **Stitching**
  - Similar Context can stich together with operators
- **Property**
  - sensedProperty
  - derivedProperty
  - storedProperty
- Operator
  - Logical Operator - Complex operators





How to Model Context

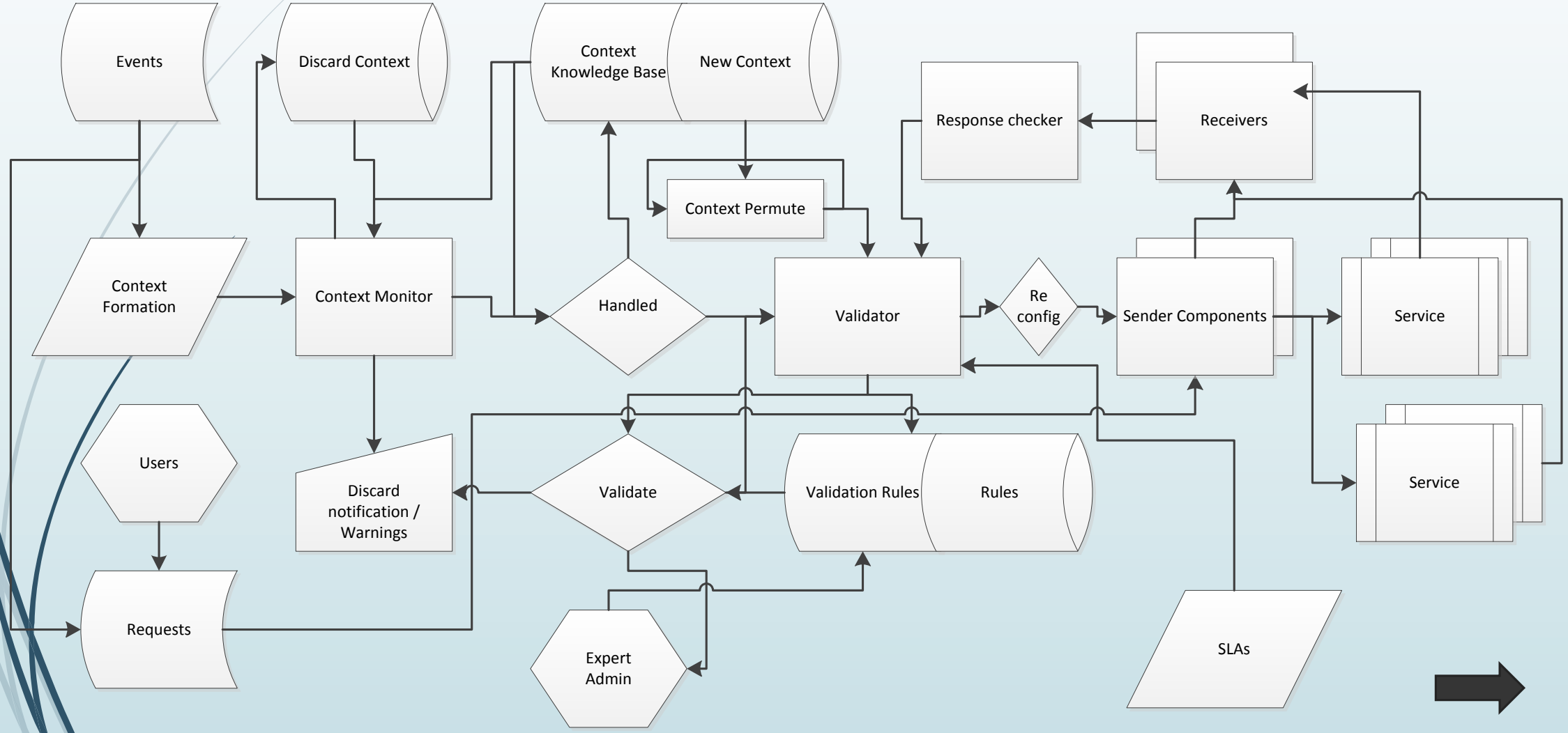
- **Propositions**
  - Context Properties, Joined together by Operator
- **Stitching**
  - Similar Context can stich together with operators
- Operator Build together the sub category into a tree like structure c-tree
- Variability
  - AS a Sensor is added it s properties are added to the leaf node
  - In al the tee structures available the properties for the other devise are tried incrementally and removed if not applicable as per the validation rule



Sub-Cat <Occupant> OP (P1, P2)
Sub-Cat <Environment> OP((P1,P2,P3), OP (P3, P4)
Sub-Cat <SC3> OP(.....)
Sub-Cat<SC4> OP(.....)
Sub-Cat<SC5> OP(.....)

OP(Sub-Cat1, Sub-Cat2, Sub-Cat3 ..... SubCatn)

Coordinator Design



Algorithm

- Generate Context
- Validate
  - when new context
  - Check for the variability
  - Add the properties
  - Prune by validation/reason rules
  - Use Recionfgurator to chose services
  - Send Notification
  - Response – Check
  - Validated - Goes to KB
  - If Not Validates – Expert
    - Ignore
    - Add to Discard
    - Add to Accept

Algorithm Context_Update (Op, Context_data)

```
Generate (context, user) based on new context and request data
For each variation in op in op set
      Add node to C-Tree
     Use the Context Knowledge base discard the nodes and connections
End For
For each sub tree in the C_Tree DFS
      Validate the C_tree with the validation rules and Prune any edges
      Accept_Set <- Null
    Do Until all Service explored in RR select service or Maximum Tries exhausted
          If Service Request fails try next
          Validate response against the response checker
         If Response Accepted
             Check Response SLA
             If the Service response within SLA response time, Add to Context_Accept set
          Else
            Discard and try next service
    End Do
    If Accept_set not Null
          Add the Accept_Set data to Context Knowledge base
      Else
          Present the Context to Expert // Expert may add the context to Discard Knowledge Base
   Remove Context from ContextNew
  END For
End
```

Evaluation

- Required
  - Context Aware services
  - CPS – Simulated architecture
  - Coordinator Designed
  - Run simulations by adding Devices/Sensors
    - Introducing properties
- Comparison to be made
  - Added context to Knowledge Base
  - Ability to resolve context I lower response time
    - With Knowledge Base and with out
- In Progress - TBC

Q & A
Final

References

- Brian Blake , Decomposing composition: Service oriented software engineers , IEEE Software Volume: 24, Issue: 6, Nov.-Dec. 2007, pp. 68-77
- Emiliano Casalicchio, Daniel A Menasc´e, Vinod Dubey, and Luca Silvestri , On Optimal service selection heuristics in service oriented architectures, Science Direct, Volume 67, Issue 8, August 2010, Pages 659-675 pages 785–798.
- Daniel A. Menascé, Hassan Gomaa, Sam Malek, and João P. Sousa George Mason University, SASSY: A Framework for Self-Architecting Service-Oriented Systems; IEEE Software, Nov.-Dec. 2011, Vol.28(6), pp.78-85
- Ewing, John, Autonomic Performance Optimization with Application To Self-architecting Software Systems , Doctoral Dissertation, Menasce, Daniel A. (advisor), Gomaa, Hassan (committee member) ; Malek, Sam (committee member) ; Nash, Stephen (committee member); George Mason Universoty, 2015
- G.H. Alférez, V. Pelechano, Achieving Autonomic Web Service Compositions with Models at Runtime, Computers and Electrical Engineering (2017), http://dx.doi.org/10.1016/j.compeleceng.2017.08.004
- Hatim Guermaha,*, Tarik Fissaaa, Hatim Hafiddia,b, Mahmoud Nassara, Abdelaziz Kriouilea, A Semantic Approach for Service Adaptation in Context-Aware Environment, International Symposium on Emerging Inter networks, Communication and Mobility, Science Direct, Procedia Computer Science 34 (2014) 587 – 592
- P. Brezillon, "Focusing on context in human-centered computing", IEEE Intelligent Systems, Vol. 18, No. 3, 2003, pp. 62-66.
- Hafiddi, Hatim & Nassar, Mahmoud & Baidouri, Hicham & Asri, Bouchra & Kriouile, Abdelaziz. (2011). A Context-aware Service Centric Approach for Service Oriented Architectures.. ICEIS 2011 - Proceedings of the 13th International Conference on Enterprise Information Systems. 3. 176-183.
- Luiz Olavo Bonino da Silva, Remco Poortinga – van Wijnen, Peter Vink, A Service-Oriented Middleware for Context-Aware Applications, Journal of Network and Computer Applications Volume 28, Issue 1, January 2005, Pages 1-18
- Hatim Hafiddi, Hicham Baidouri, Mahmoud Nassar and Abdelaziz Kriouile, Context-Awareness for Service Oriented Systems,
- A.K. Dey, Understanding and using context, Personal and Ubiquitous Computing 5(1) (2001), 4–7.
-]M. Perttunen, J. Riekki, and O. Lassila, Context representation and reasoning in pervasive computing: A review, International Journal of Multimedia and Ubiquitous Engineering 4(4) (2009), 1–28.
- M.H. Tran, J. Han, and A. Colman, Social context: Supporting interaction awareness in ubiquitous environments, in: Proc. of the Int. Conf. on Mobile and Ubiquitous Systems: Networking & Services, July 2009, pp. 1–10.
- Tsigkanos, C., Kehrer, T. & Ghezzi, Architecting Smart Cyber Physical Spaces, Computing (2016) 98: 1011. doi:10.1007/s00607-016-0509-6