

# Self Introduction

---

Muhammad Taimoor Khan

M.Sc. Advanced Distributed Systems - Distinction

University of Leicester, UK

# Outline for the Presentation

---

- My personal background
- My experience
  - My software skills
  - My research
- My MSc dissertation
- PhD topic

# My Personal Background

---

**Name** Muhammad Taimoor Khan

**Date of Birth** April 05, 1978

## **EDUCATION**

Jan. 07 to Jun. 08 M.Sc. Adv. Distributed Systems (Distinction)  
University of Leicester, UK

Dec. 98 to Jun. 00 M.Sc. Computer Science (1<sup>st</sup> Class)  
Islamia University Bahawalpur, Pakistan

Oct. 95 to Oct. 97 B.Sc. Pure Mathematics , Applied Mathematics  
and Physics (1<sup>st</sup> Class)  
University of the Punjab, Lahore, Pakistan

# My Experience

---

## WORK EXPERIENCE

- Jul. 02 to Apr. 05 Software Architect/J2EE Developer  
Acrologix (Pvt.) Ltd. Lahore, Pakistan
- Jun. 00 to Jul. 02 Lead Java Design and Development Team  
WaxSys (Pvt.) Ltd. Lahore, Pakistan

## TEACHING

- Oct. 05 to Date Assistant Professor (leave 01/07 to 06/08)  
COMSATS Institute of Information Technology,  
Abbottabad, Pakistan
- Oct. 07 to Jun. 08 Teaching/Tutorial Assistant  
School of Mathematics and Computer Science,  
University of Leicester, UK
- Apr. 05 to Oct. 05 Lecturer  
Baluchistan University of IT and Management  
Sciences, Quetta, Pakistan
- Oct. 03 to Apr. 05 Research Assistant  
University of the Punjab, Lahore, Pakistan

# My Software Skills

---

<b>Languages</b>	Java (J2EE), C/C++, XML, RDF, OWL, WML, WML Script, Microsoft embedded VB/VC, Water 2.0, Promela
<b>Architectures</b>	EJB (1.x/2.x/3.x), CORBA/ RMI-IIOP, Web Services
<b>Modeling</b>	RUP 5.5, UML 1.x/2.x, CommUnity, SRML (Sensoria Reference Markup Language)
<b>Frameworks</b>	JNI, RPC, J2EE (JSP/Servlets/ Design Patterns), WAP, J2ME, Web Services
<b>Concepts</b>	OOA/OOD (Design Patterns, Refactoring)
<b>Databases</b>	MS SQL Server 2000/2003
<b>Servers</b>	Web Logic 6.x/7.x, JRun 3.x, Tomcat 4.x-6.x, IIS, IBM Web Sphere 5.x, JBoss 3.x
<b>Toolkits</b>	JDK, Nokia WAP Toolkit, Open wave SDK/Push Library/Gateway
<b>Tools</b>	Borland JBuilder 9.x/10.x/2006, Rational Rose 98/2K, MS Visio 2002, Lotus Notes, OPNET 8.x-14.x, Eclipse 3.x, Spin 3.x/4.x, Altova UModel 2008
<b>Systems</b>	Windows (2K), Windows 2000 Server, Linux, Windows CE 3.0, SavaJe 2.0
<b>Note:</b>	Java (08 years experience) C/C++ (03 years experience)

# My Research

---

## **Feb. 08 to Jun. 08      M.Sc. Dissertation**

- Formalize the NASA's specification document
- Modeling in UML Model Checker
- Verifying certain protocol properties presented in state-machines

## **Oct. 05 to date      Focused in exploring**

- Fault tolerant scheduling in Computational Grids
- Design of middleware for pervasive systems
- Information Organization in smart spaces
- Interoperability issues in pervasive multimedia networks
- Project **UbiCampus**

## **Oct. 02 to Apr. 05      To design and implement**

- QoS aware MAC protocol for IEEE 802.11 WLANs
- Multi-channel framework for web services based business
- Model users using ANN for location based pervasive applications
- Framework for pervasive e-business

# **My Dissertation: Space Link Extension - Service Management (SLE-SM)**

---

Supervised By : Prof. Reiko Heckel  
Student Name : Muhammad Taimoor Khan  
Student ID : 079001210  
CFS User ID : mtk6  
Email ID : mtk6@leicester.ac.uk  
Second Marker : Emilio Tuosto

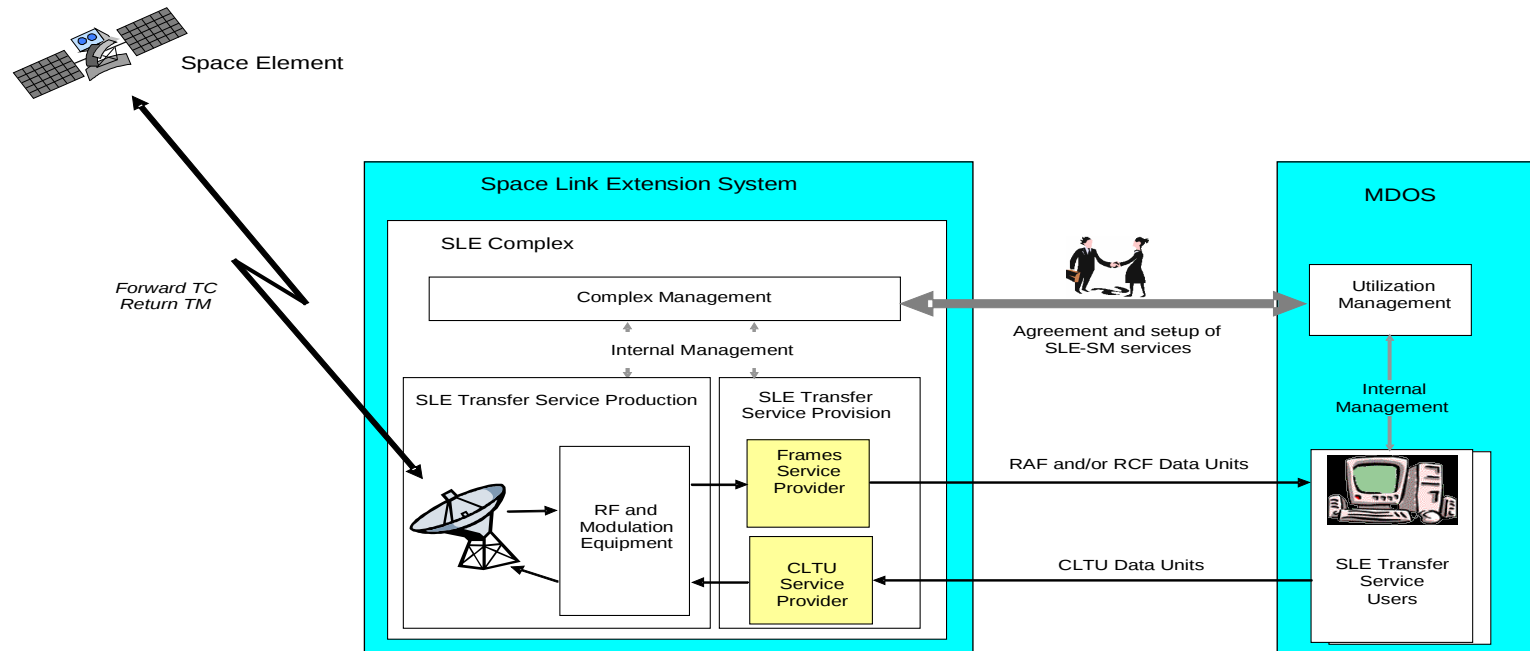
# Outline

---

- + Introduction
- + Methodology
- + Technical Background
  - + Unified Modelling Language - Metamodel
  - + XML Metadata Interchange
  - + UML Model Checker (UMC)
- + Case Study-I (Three-phase Procedure Pattern)
- + Case Study-II (Service Package service)
  - + Informal Model
  - + Technical Inconsistencies
  - + Proposed Model
  - + Analysis of the proposed model
- + XMI2UMC Transformer
  - + Introduction
  - + Transformation Rules
- + Conclusion



# Space Link Extension - Service Management (SLE-SM)



# Methodology

---

## Requirements

- ❑ NASA specification document
  - ❑ descriptive requirements for communication protocols
  - ❑ state diagrams based model for the requirements

## Problem

- ❑ Issues with the specification document
  - ❑ described requirements vs state diagrams - inconsistency
  - ❑ state diagram model itself - inconsistency + informality

## Solution Design

- ❑ formalise the given model - state machines
- ❑ analyse the model - assumptions/limitations
- ❑ verify the proposed model using CTL- consistency
- ❑ develop the XMI to UMC transformer - compatibility

# Unified Modelling Language - Metamodel

---

- + Complete modelling language
- + Strong in terms of modularity
- + UML Superstructure
- + State Machine Model

# XML Metadata Interchange (XMI)

---

- Defining framework between XML Data and Objects for;
  - Integration
  - Interchanging
  - Manipulation
- Used to interchange UML models
- Used to publish design metadata on the web

# UML Model Checker (UMC)

---

- ❑ Project by CNR, Italy (Mazzanti and colleagues)
- ❑ Testing the appropriateness of UML methodology
- ❑ Study the state and event oriented behaviours (using temporal logic (CTL))
- ❑ UMC Model = Templates + Objects

# Case Study-I (Three-phase Operation Procedure)

---

## □ Phases

- Invocation
- Acknowledgement
- Failed/Successful Return

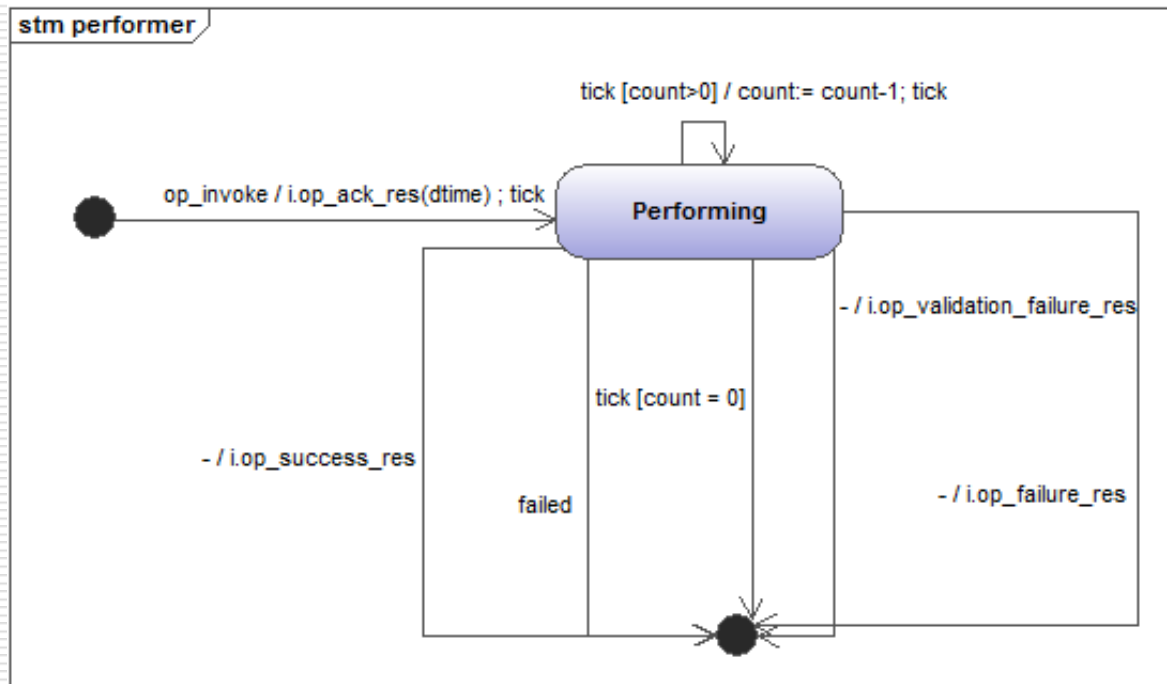
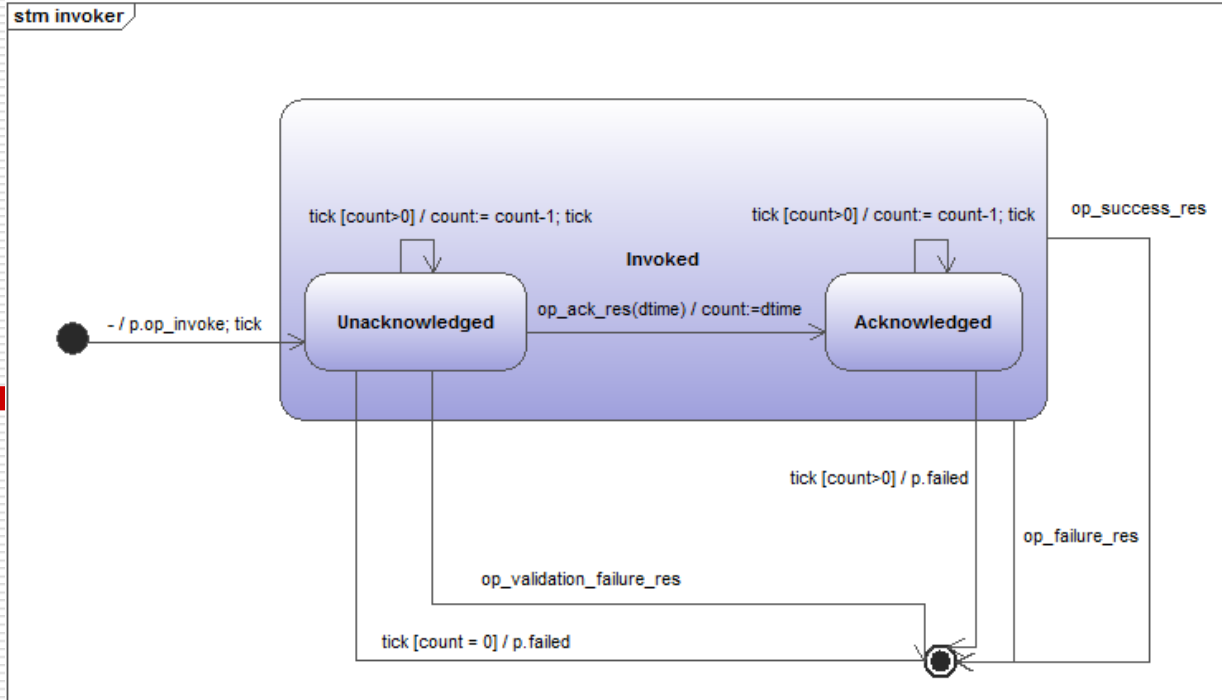


# Selected Informalities in modelled system

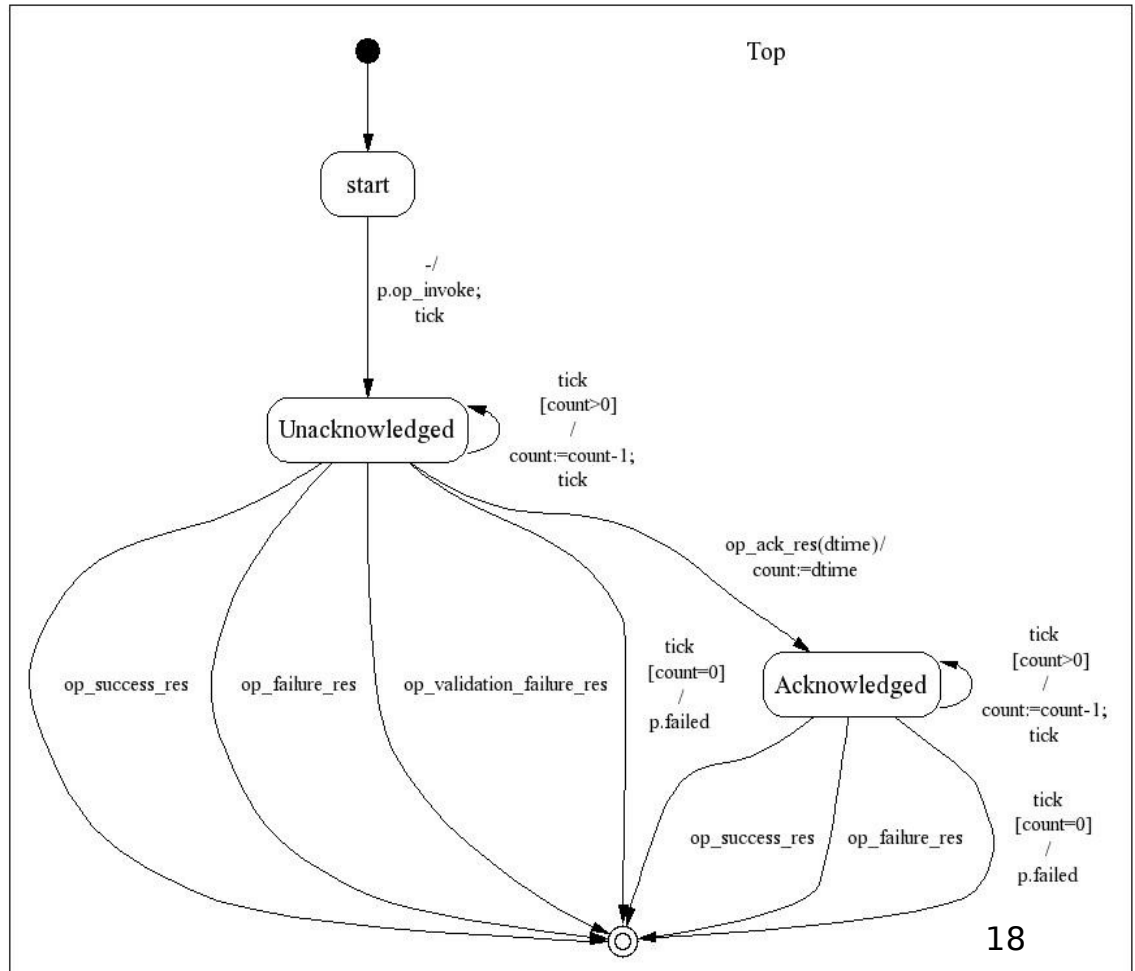
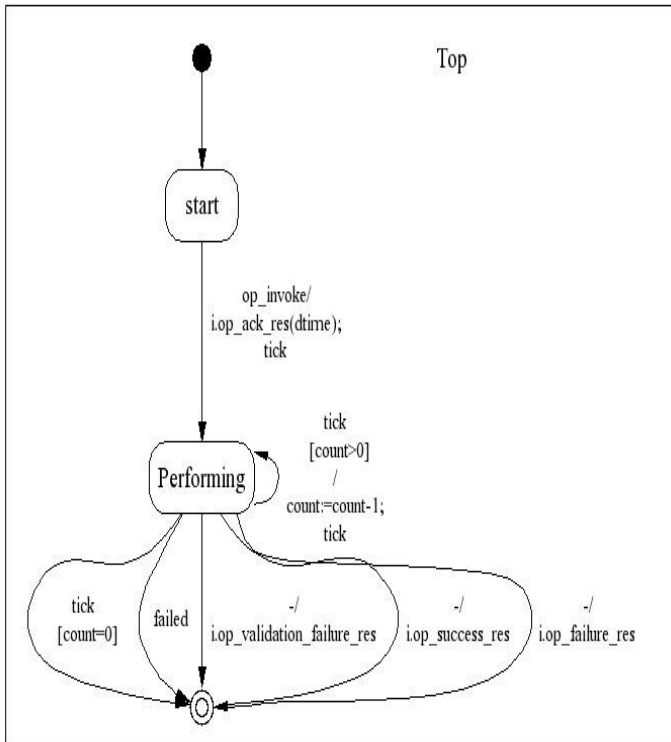
---

- Informal modelling
  - Improper labeled e.g. no difference in `failed` and `op_FR`
- Missing behaviour
  - No `timeout` for Performer
- Ambiguous Semantics
  - `op_FR` and `failed` both are failure messages
- Missing Semantics
  - What about `un-delivered`, `replayed` and `lost` messages





# UMC Model



# Analysis of the Model

---

- Assumptions
  - Infinite System definition (e.g. counter)
- Verifiable Properties
  - Absence of Deadlock
    - “*AF FINAL*” returns *true*
  - Absence of Context Inconsistency
    - “*EF (my\_invoker1.count=0 AND my\_performer1.count>0)*” returns *true*

# Case Study-II

## Service Management service

stat

---

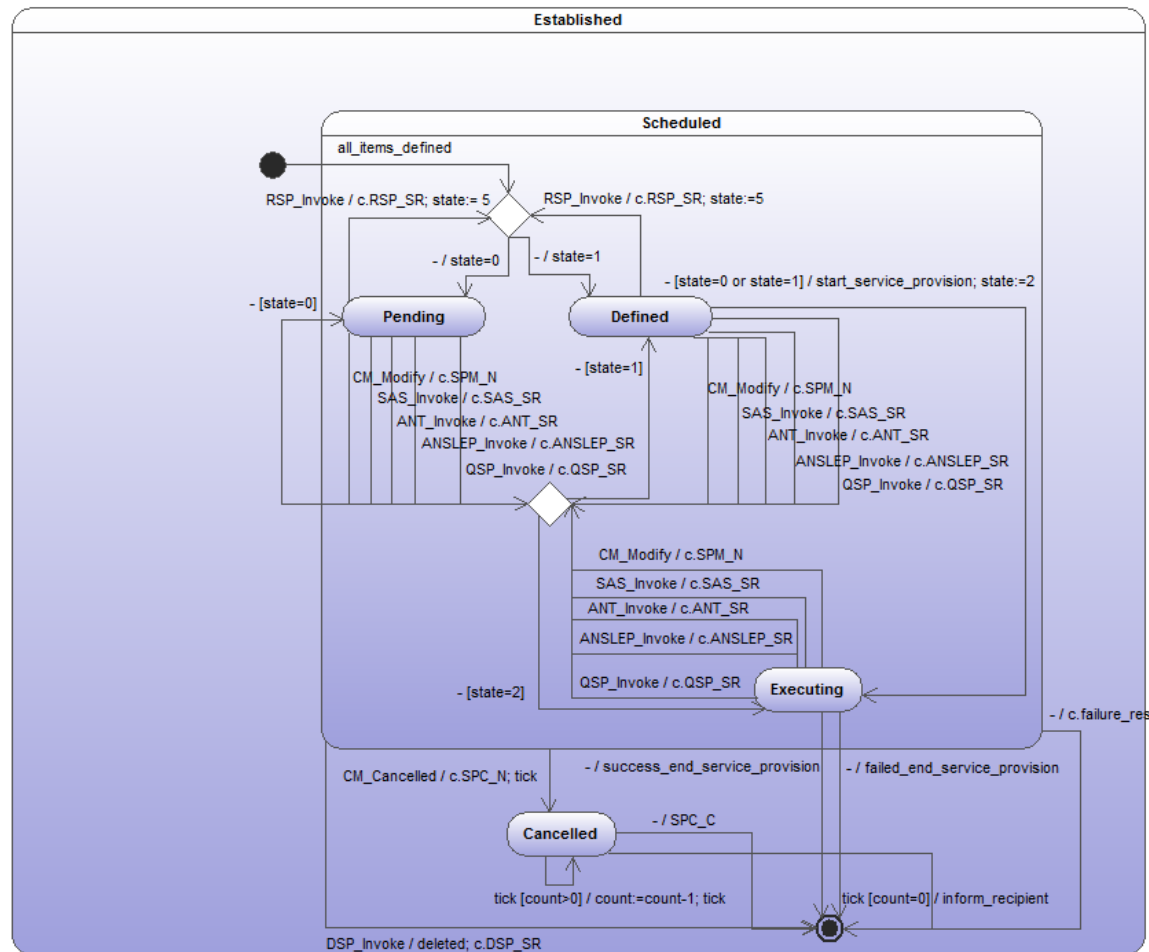
# Selected Informalities in modelled system

---

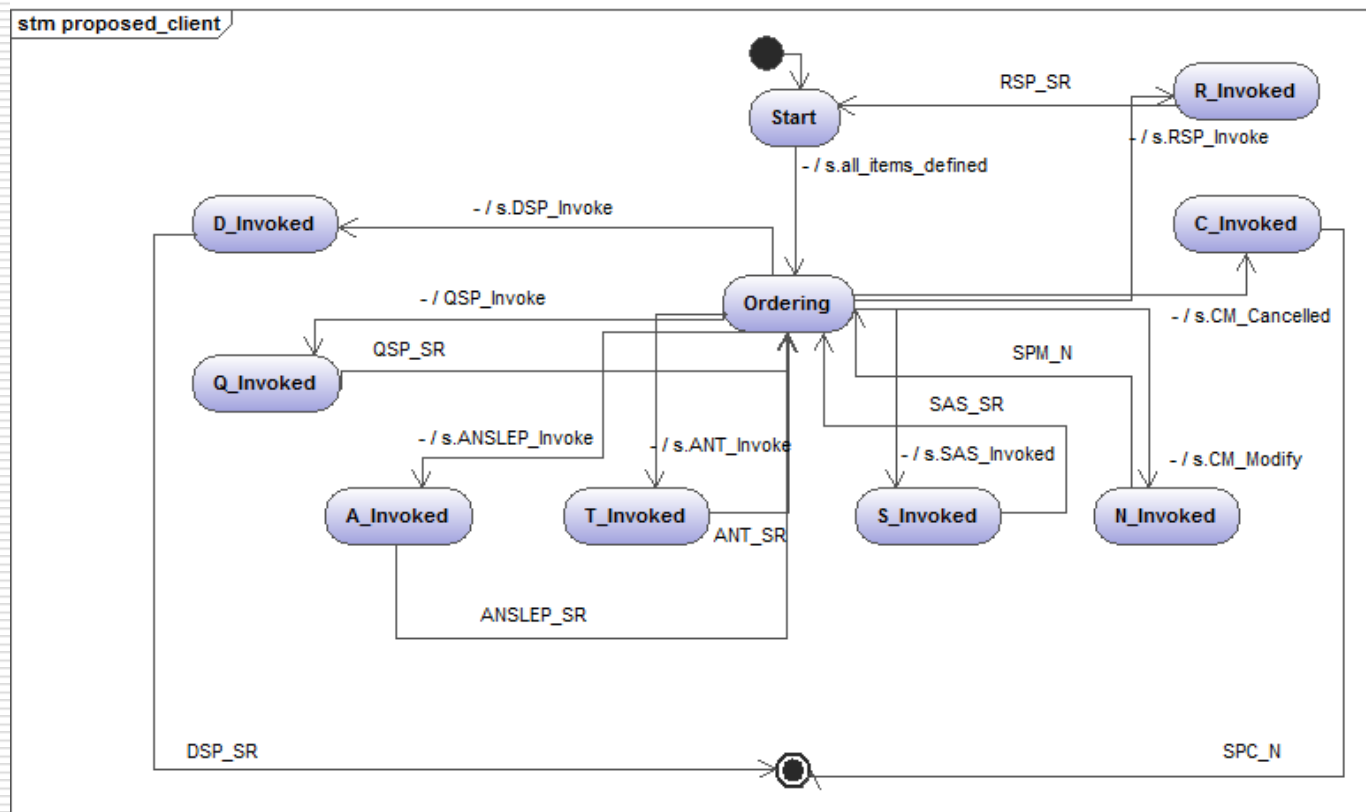
- Informal modelling
  - No clue about the execution of `all_items_checked`
- Missing behaviour
  - Counter behaviour for `QSP_SR`
- Ambiguous Semantics
  - No check about the termination of the service
- Missing Semantics
  - Entire counter behaviour is missing

# Proposed Model (Service Package Server)

stm service\_pkg



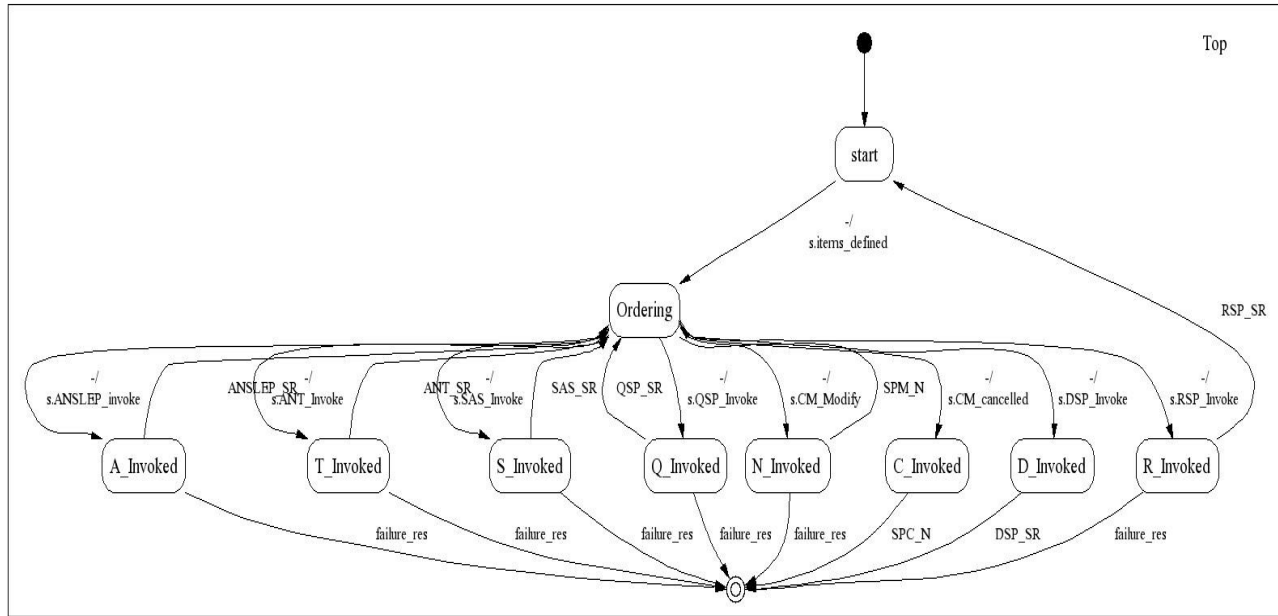
# Proposed Model (Service Client)







# UMC Model



# Analysis of the Model

---

- Verifiable properties
  - Deadlock
    - “*AF FINAL*” returns *true*
  - Alternative (non-reliable communication)
    - Deadlock avoidance

# XMI2UMC Transformer

---

- Command-line and web service (versions)
  - Core Classes
  - Persistence Classes
  - Utility Classes
- Transformation Rules
- Support for different XMI and UML versions
- Supported State Diagram Features

# Conclusion and Future Work

---

- ❑ Identification of informalities and inconsistencies
- ❑ Formalised State Machines
- ❑ XMI2UMC Transformer
- ❑ UMC model
- ❑ Verifiable behaviour

# PhD Topic

---

- introduce formal methods applications to computer algebra
- equip computer algebra system – formal specification language
- integration of computation and reasoning
- theorem provers with computing capabilities
- tool for computer algebra languages

---

Thanks!!  
Questions ???