Course Code: **CSE-869**

Title: **Model Driven Software Engineering (MDSE)**

Credit Hours: **3+0**

Objectives

1. Provide students with the overview of Model Driven Software Engineering (MDSE) and Model Driven Architecture (MDA).
2. Prepare students for further specialized studies in MDSE, in particular for the MS and PhD thesis.
3. Prepare students to undertake key research activities such as literature review, research paper reviews and teaches them how to present their own research work in the form of research paper.
4. Familiarize the students with the state of art Modeling technologies and languages such as Object Constraints Language (OCL), Action Language for FUML (ALF).
5. Introduce students to the importance of MDA and automated model transformation.

Outcomes

6. Students will be able to model a given system using object oriented models and capture the constraints on the models using OCL.
7. Students will be able to understand the UML meta model and be able to develop their own domain specific languages (DLS's) or UML Profiles for modeling.
8. Students will be able to perform basic transformations in languages, such as kermeta, ATL, QVT, etc, to automatically transform one meta model into another.
9. Students will be able to study and understand the state of the art papers from leading conference and journals in the domain.
10. Students will be able to review the existing literature and formulate problems that may be solved during their MS or PhD thesis.

Contents

11. Introduction to Model Driven Software Engineering - MDSE Overview:-
    a. Modelling Modern Software Systems

b. MDE Basics - Modeling Object Oriented Systems using UML Class diagram

12. **MDSE Basics -** Modeling system behaviour with Interaction diagrams Modeling system constraints using Object Constraints Language (OCL) Modeling dynamic system behaviour with UML State machines:-

a. **Advanced Behavioural Modeling**. Adding guards, triggers, and effects - Concurrent state machines.
b. Understanding UML Meta-Model
c. Meta-model- Basics, Why Meta-model?
d. Meta-model of UML Class diagram
e. Creating Domain specific Models
f. Creating domain specific languages - DSL and UML Profiles
g. From DSL to code generation - tools and techniques
h. Meta-modeling in MDA
i. Defining modelling languages – meta-modeling approach
j. UML as a meta-model based language - A walk through the UML meta-model
k. Presentation / Guest Lecture by Domain expert
l. Model Transformation
m. Model Transformations - From one model to another
n. Model Transformations - Model to Model Transformation - QVT
o. Model Transformations- Kermeta, ATL, Java
p. Selecting a transformation language for your project
q. Case Studies
r. Transforming OO model to Relational Model
s. Model Transformations - Model to Code Transformation
t. Model to Text transformation issues
u. Empirical evaluation / Experiments (Model transformation languages)
v. Guest Lecture: Modelling Real time systems - state of the art and research challenges
w. Evaluation / Discussion on Model transformation (State of the art)
x. Evaluation/Discussion on Model transformation (State of the art)-Part II
y. Understanding semantics - Arriving at Executable Models

z. Adding Semantics to the models: A deep drive through UML semantics

aa. From Models to Executable Models

bb. Executable Models: Foundation UML and Action Languages - ALF, ASL

cc. Executable Models: Foundation UML and Action Languages - ALF, ASL. Cont

dd. Foundational UML

ee. Systematic Review of OMG® Specifications

ff. Presentation / Guest Lecture by Domain expert

gg. Lectures: Following is the non exhaustive list of OMG® Specifications. Some of these may be included based on the available expertise: -

(1) Production Rule Representation,

(2) Business Process Model and Notation

(3) Workflow Management Facility

(4) CORBA Component Model

(5) Real-time CORBA

(6) IDL to Java, Java to IDL,

(7) OMG Systems Modeling Language

(8) Ontology Definition Metamodel

(9) Service oriented architecture Modeling Language

(10) Software Process Engineering Metamodel

(11) UML Profile for MARTE: Modeling and Analysis of Real-time and Embedded Systems

(12) UML Profile for System on a Chip

(13) UML Profile for Software Radio (aka PIM & PSM for Software Radio Components)

(14) Knowledge Discovery Metamodel

(15) PIM and PSM for Smart Antenna

(16) Unified Profile for DoDAF and MODAF

(17) Robotic Technology Component

hh. Overview of Current Research Areas in MDSE

ii. Advance Topics in MDSE

jj. Research Presentations

13. **Library**

a. DSL Engineering: Designing, Implementing and Using Domain-Specific Languages, Markus Voelter, Create Space Publishing 13

b. Model-Driven Software Engineering in Practice, by Marco Brambilla, Jordi Cabot, Manuel Wimmer, Morgan & Claypool, 12

c. Model-Driven Software Development: Technology, Engineering, Management, by Thomas Stahl and Markus Voelter, Wiley, 2006

d. MDA Explained: The Model Driven Architecture : Practice and Promise, by Jos B. Warmer, Addisson-Wessely Professional, 2003

e. Executable UML: A Foundation for Model-driven Architecture, S Mellor and M. Balcer

14. Benefits and potential research areas to the society

15. **Benefits**. Here are some of the benefits of MDSE to the society:-

a. The core MDSE outcome is the verification of system design in early development phases. Therefore, it is highly supportive to meet the business objectives like productivity and time-to-market.

b. MDSE significantly reduces the development efforts, particularly for embedded systems, due to its early design verification capabilities. Consequently, it has been frequently utilized in various industrial and research practices for the development of embedded systems [1-13].

c. As the structure, behaviour and constraints of the system are captured within a single model, any corrective, perfective or adaptive changes can be accommodated by performing the corresponding alterations in the model only. Therefore, it is not required to perform such changes in each software development phase. Consequently, the maintenance efforts are notably reduced.

d. MDSE enhances the reusability as the developed meta-models and their transformation techniques can be applied to the different systems of a similar business domain.

e. MDSE has the potential to raise application development to the next level of abstraction. Models hide ‘plumbing’ details, while highlighting the important concepts and essential business logic. This aids in understanding and communicating intentions among the various stakeholders in a language, which user understands.

f. To summaries, MDSE has a potential to mitigate the risk of software projects failure, resultantly the investment; as the requirements are rapidly transformed to design (model) which are validated in the early software development stages.

16. **Research Areas**. As MDSE incorporates vast range of software engineering aspects, there is variety of research opportunities for practitioners and researchers of the domain. However, the most important research directions can be summarized as follows: -

a. Modelling: It is always challenging to select appropriate UML profiles and their diagrams / notations for the specification of the system requirements. Therefore, one appealing research direction is to explore the UML profiles (e.g. FUML and SYSML etc) and propose the best solution by harnessing the strengths of different UML profiles. This should be done by considering the complete structure, behaviour and constraint requirements of the large and complex systems. To be more realistic, this research can be done by considering the particular characteristics of different systems separately e.g. embedded systems [1-13], data base systems etc.

b. Model Transformation: It is demanding task to ensure the correctness of model transformation to get the desired output model. Therefore, it is the actual demand of the industry to explore and develop the generic mechanism to ensure the correctness of both M2M and M2T transformations.

17. **Relevant Research Journals/Conferences:** MDSE is a growing area. It is evident as following reputed journals and conferences are focusing in this area: -

a. MODELS Conference
b. ECMFA Conference
c. ICMT Conference
d. MODELSWARD Conference
e. TTC part of STAF conferences
f. Special issue of MDSD on EJIS Journal
g. Sosym Journal Springer

18. **References**

a. Giuseppe Di Guglielmo, Luigi Di Guglielmo, Andreas Foltinek,, Masahiro Fujita, Franco Fummi, Cristina Marconcini and Graziano Pravadelli: **On the integration of model-driven design and dynamic assertion-based verification for embedded software**, Journal of Systems and Software, Volume 86, Issue 8, August 2013, Pages 2013–2033. DOI: 10.1016/j.jss.2012.08.061

b. Ermeson Andrade, Paulo Maciel, Gustavo Callou and Bruno Nogueira: **A Methodology for Mapping SysML Activity Diagram to Time Petri Net for Requirement Validation of Embedded Real-Time Systems with Energy Constraints**, Third International Conference on Digital Society ICDS 2009, Pages 266-271, DOI: 10.1109/ICDS.2009.19

c. Imran R. Quadri, Etienne Brosse, Ian Gray, Nicholas Matragkas, Leandro Soares Indrusiak, Matteo Rossi, Alessandra Bagnato and Andrey Sadovykh: **MADES FP7 EU Project: Effective High Level SysML/MARTE Methodology for Real-Time and Embedded Avionics Systems**, 7th International Workshop Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC) 2012, Pages 1-8, DOI: 10.1109/ReCoSoC.2012.6322882

d. Jorgiano Vidal, Florent de Lamotte, Guy Gogniat, Philippe Soulard and Jean-Philippe Diguet: **A co-design approach for embedded system modeling and code generation with UML and MARTE**, Conference and Exhibition Design Automation and Test in Europe (DATE) 2009, Pages 226-231, DOI: 10.1109/DATE.2009.5090662

e. Fernando Herrera , Hector Posadas, Pablo Penil, Eugenio Villar, Francisco Ferrero, Raul Valencia and Gianluca Palermo: **The COMPLEX methodology for UML/MARTE Modeling and design space exploration of embedded systems**, Journal of Systems Architecture, Elsevier 2014, Volume 60, Issue 1, Pages 55-78, DOI: 10.1016/j.sysarc.2013.10.003

f. Stephane Lecomte, Samuel Guillouard, Christophe Moy, Pierre Leray and Philippe Soulard : **A co-design methodology based on model driven architecture for real time embedded systems**, Mathematical

and Computer Modelling, Elsevier 2011, Volume 53, Issue 3-4, Pages 471-484, DOI: 10.1016/j.mcm.2010.03.035

g. DeJiu Chen, Lei Feng, Tahir Naseer Qureshi, Henrik Lonn and Frank Hagl : **An architectural approach to the analysis, verification and validation of software intensive embedded systems**, Computing, SPRINGER 2013, Volume 95, Issue 8, Pages 649-688, DOI: 10.1007/s00607-013-0314-4

h. Elvinia Riccobene and Patrizia Scandurra: **Integrating the SysML and the SystemC-UML profiles in a model-driven embedded system design flow**, Design Automation for Embedded Systems, SPRINGER 2012, Volume 16, Issue 3, Pages 53-91, DOI: 10.1007/s10617-012-9097-7

i. Huascar Espinoza, Daniela Cancila, Bran Selic and Sebastien Gerard: **Challenges in Combining SysML and MARTE for Model-Based Design of Embedded Systems**, LNCS SPRINGER 2009, Volume 5562, Page 98-113 DOI: 10.1007/978-3-642-02674-4_8

j. Luciane Telinski Wiedermann anger, Inali Wisniewski Soares and Paulo Cezar Stadzisz: **A Brazilian survey on UML and model-driven practices for embedded software development**, The Journal of systems and software 2013, Pages 997-1005

k. Ning Ge, Marc Pantel and Xavier Cregut: **Formal Specification and Verification of Task Time Constraints for Real-Time Systems**, LNCS SPRINGER 2012, Volume 7610, Pages 143-157 DOI: 10.1007/978-3-642-34032-1_16

l. Luciano Baresi, Gundula Blohm, Dimitrios S. Kolovos, Nicholas Matragkas, Alfredo Motta, Richard F. Paige, Alek Radjenovic and Matteo Rossi: **Formal verification and validation of embedded systems: the UML-based MADES approach**, Software and Systems Modeling SPRINGER 2013, DOI: 10.1007/s10270-013-0330-z

m. Minh Chau Nguyen, Eunkyoung Jee, Jinho Choi and Doo-Hwan Bae: **Automatic Construction of Timing Diagrams from UML/MARTE Models for Real-Time Embedded Software**, Proceedings of the 29th Annual ACM Symposium 2014, Pages 1140-1145, DOI: 10.1145/2554850.2555011

n. OMG, M2M/Operational QVT Language, 2007.
o. Z.Drey, C.Faucher, F.Fleurey, V.Mahe, D.Vojtisek, Kermeta Language Reference Manual, https://www.kermeta.org/ docs/KerMeta-Manual.pdf,April2009.
p. Eclipse ATL, Last Accessed August 2014. www.eclipse.org/atl/
q. Oldevik, J.: MOFScript user guide. Version 0.6 (MOFScript v 1.1. 11) (2006)
r. Eclipse Acceleo, Last Accessed August 2014. https://wiki.eclipse.org/Acceleo/
s. Apache Software Foundation. (2010, Nov.) The Apache Velocity Project. [Online]. Available: http://velocity.apache.org/