UNIVERSITÄT SIEGEN

Co-funded by the Erasmus+ Programme of the European Union

Erasmus+ SkoPS

Intellectual Output 6 - Cyberphysical Real-Time Systems

Project Title	Empowering the European Workforce Development through Online/Virtual Skills Training for Digital Transformation toward Mitigating the Impact of Pandemic Situations (SkoPS)		
Project Acronym	SkoPS	**Project Number**	2020-1-DE01-KA226-HE-005772
Date	2022-10-30	**Deliverable No.**	D0.6
Contact Person	Bahareh Kiamanesh Ali Behravan	**Organization**	USI
Phone		**E-Mail**	Bahareh.Kiamanesh@uni-siegen.de Ali.Behravan@uni-siegen.de
Version	1.1	**Confidentiality level**	Public

UNIVERSITÄT SIEGEN Petanux Universitat d'Alacant European Institute for Local Development

Version History

Version No.	Date	Change	Editor(s)
1.0	03.11.2021	Initial Version	Bahareh Kiamanesh
1.1	30.10.2022	Final Version	Bahareh Kiamanesh

Contributors

Name	Organization
Bahareh Kiamanesh	USI
Ali Behravan	USI

Disclaimer

This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Table of Contents

Introduction

Course Plan Template can help the partners to prepare the course materials that help us to organize the materials and contents of courses in the SkoPS project in standard form. Some other Forms and templates for the student assessment Plan and Evaluation Questionaries Plan for each Course can be added to this Template.

1.1 Abstract

In Real-time systems, sensors collect data in periodic intervals, and the system should respond via actuators. The real-time program should interact with the real-world environment and execute in a system with defined resources. There should be time bounds when the system responses should be delivered in. The ability of the system to respond to timing and computational demands depends on the system's requirements. For the close-relation events, the system should make a schedule to compute and respond to them in desired time. During system operations and computations, failures are possible to occur and have various consequences. These consequences may be distinguished as minor when the effects are not considerable and otherwise are a catastrophe.

1.2 Purpose of the document

The purpose of this document is to have the same template for all courses. In this template, for each course, some features are considered.

1.3 Relation to other deliverables

This Report is part of the D0.6, which is the general structure of each course.

Template

Course Plan Template

Course ID and Title:	Intellectual Output (IO) 6 - Cyber-Physical Real-Time Systems		
Course Duration:	2/3 months	**Course ECTS:**	1/1.5
Leading Organization:	University of Siegen (USI)		
Course Media:	Text / Text File / Video / Website/ Internet		
Laboratory (Yes/No)	No		

Course Description:

This module concentrates on the system aspects of embedded real-time systems. It provides the central requirement and materials, such as introducing, modeling, and designing the embedded control systems represented in 8 weeks. The primary purposes of the course are to explain the application fields of embedded control systems and real-time embedded systems to understand methods to develop embedded control systems such as modeling, design, and analysis. Students will know various paradigms and design principles for real-time embedded systems. In addition, the fundamentals of embedded real-time systems are intended for knowledge gained from new developments (e.g., the Internet of Things(IoT)) to provide the basis for research activities in the field of cyber-physical systems.

The course "Cyberphysical Real-Time Systems" and "Embedded Control Systems" will enable the students to:

- Understand the basics of the real-time embedded systems
- Understand the typical components of the real-time embedded system
- To learn the design process of the real-time embedded system applications
- To understand different communication interfaces

Course Materials and Equipment (Prerequisite)

Not Required

Teaching and Learning Activities

- Revision and advancement of existing teaching/learning materials
- Type: Open Educational Resource (OER)
- Level: Industrial workers and bachelor's students, Master's degree and PhD
- Primary learning format: Online teaching
- Reading materials: Word file and Slides
- Discussion and interactive activities
- Webinars: Some webinars to provide the materials will hold at the end of each course
- Videos: 2-3 Videos for each chapter will be prepared to present the course outlines and students' motivation
- Quiz: To analyze the students after each chapter, a quiz with multiple-choice questions or short-answer questions will be designed

Course activities:

- Revision and advancement of existing teaching/learning materials

- Discussion and interactive activities
- Evaluation Activities

Course Objectives:

Intended Module Learning Outcomes and Qualification Goals:

On completion of the module:

- Students know about requirements, paradigms, concepts, platforms, and models of cyber-physical systems.
- Students understand the non-functional properties of embedded systems and concepts and methods for real-time capability and fault tolerance.
- Students should become familiar with various components and design principles to translate them into concrete systems for solving application problems.
- Students can judge opposing design approaches (such as time-triggered and event-triggered control) and adapt these to new application problems.
- Students can also select platform technologies such as communication protocols, processors, and operating systems and judge their suitability for real-time, security, and reliability requirements.

Table of Contents

1. Introduction to Cyber-physical Real-Time Systems

1.1 Embedded Systems

1.1.1 Cyber-physical system (CPS)

1.1.2 Applications

1.1.3 Example

1.1.4 Design Requirements of Cyberphysical Real-Time Systems

1.2 The Design Process

1.2.1 Modeling

1.2.2 Design

1.2.3 Analysis

2. Modeling and Mathematical Descriptions of Dynamic Systems (Continuous Dynamics)

2.1 Mathematical Descriptions of Dynamic Systems

2.2 Continuous Dynamics

2.2.1 Newtonian Mechanics

2.2.2 Actor Models

2.2.3 Properties of Systems

2.2.4 Causality

2.2.5 Memory lessness

2.2.6 Linearity and Time Invariance

2.2.7 Stability

2.3 Feedback Control

2.3.1 Feedback Composition

3. Modeling and Mathematical Descriptions of Dynamic Systems(Discrete Dynamics)

3.1 Discrete Systems

3.2 The Notion of State

3.3 Finite-State Machines (FSMs)

3.3.1 Transitions

3.3.2 Transitions-Visual Notation

3.3.3 Logical operators

3.3.4 FMS, Action, and Guard

3.3.5 Examples of valid Actions

3.3.6 One Example of Discrete Dynamics

3.4 Update Function

3.5 Extended State Machines

3.6 Determinacy and Receptiveness

3.7 Nondeterminism

3.7.1 Example of the Nondeterminism

3.7.2 A Formal Model for Nondeterminism

3.8 Behaviors and Traces

3.9 A Computation Tree for FMS

4. Modeling and Mathematical Descriptions of Dynamic Systems (Hybrid Systems)

4.1 Modal Models

4.2 Actor Model

4.3 Modal Model and Continuous Inputs

4.4 Hybrid Systems: Modal Model and State Refinements

4.5 Classes of Hybrid Systems

4.6 Timed Automata

4.7 Control System

4.8 Supervisory Control

5. Modeling and Mathematical Descriptions of Dynamic Systems (Composition of State Machines)

5.1 Concurrent Composition

5.1.1 Side-by-Side Synchronous Composition

5.1.1.1 Properties of Synchronous Composition

5.1.2 Side-by-Side Asynchronous Composition

5.1.2.1 Properties of Side-by-Side Asynchronous Composition

5.1.3 Asynchronous Composition for Interleaved Semantics

5.1.4 Cascade Composition (Serial Composition)

5.1.4.1 Properties and Requirements of Cascade Composition

5.1.5 General Composition

5.2 Hierarchical State Machines

6. Modeling and Mathematical Descriptions of Dynamic Systems (Concurrent Models of Computation)

6.1 Structure of Models

6.2 Models of Computation (MoC)

6.3 Example of Clock Signal

6.4 Actor Networks as a System of Equations

6.5 Fixed-Point Semantics

6.6 Synchronous-Reactive Models

6.7 Actors With Feedback

6.8 Feedback Models

6.8.1 Feedback Models Example

6.9 Constructive Semantics: Multiple Signals

6.10 Constructive Semantics: Multiple Actors

6.11 Synchronous-Reactive Languages

6.12 Dataflow Models of Computation

6.13 Dataflow Principles

6.14 Scale Actor

6.15 Actor Delay

6.16 FSM model for the Delay actor

6.17 Asynchronous Firing Actor and Buffer

6.18 Problems in Dataflow

6.19 Cascade Composition of Dataflow Actors

6.20 Synchronous Dataflow SDF

6.21 SDF Actors with Tokens

6.22 Balance Equations
6.23 Synchronous Dataflow
6.24 Schedule Example
6.25 Compactly Representing the Balance Equations
6.26 Dynamic Dataflow
6.27 Model of Computation
6.28 Timed Models of Computation
6.29 Time-Triggered Models
6.30 Discrete Event Systems
6.31 Continuous-Time Systems
6.32 Petri Nets
6.33 Example for Petri Nets

7. Design of Embedded Control Systems

7.1 Introduction to Distributed Real-Time Systems

7.1.1 Introduction to Embedded Real-Time Systems
7.1.2 Embedded systems vs. Real-Time Computing
7.1.3 Cyberphysical Systems
7.1.4 Characteristics of Real-Time Systems
7.1.5 Functional Requirements
7.1.6 Temporal Requirements
7.1.7 Control System
7.1.8 Latency Jitter
7.1.9 Distributed Control System
7.1.10 Fail-Safe versus Fail Operational
7.1.11 Classification of Real-Time Systems
7.1.12 Guaranteed Timeliness Systems
7.1.13 Best Effort Systems
7.1.14 Resource Adequacy
7.1.15 Predictability in Rare Event Situations
7.1.16 The notion of the State and Event
7.1.17 Time-Triggered (TT) vs. Event-Triggered (ET)
7.1.18 Hard Real-Time versus Soft Real Time
7.1.19 Temporal Requirements
7.1.20 Jitter at the Application Level
7.1.21 Logical versus Temporal Control

7.2 Modeling of Distributed Real-Time Systems

7.2.1 Model Construction
7.2.2 Assumption Coverage
7.2.3 Load and Fault Hypothesis
7.2.4 Elements of a Real-Time System Model
7.2.5 Structure Overview
7.2.6 Process
7.2.6.1 Simple Process (SP)
7.2.6.2 Complex Process (CP)
7.2.7 Model of an S Process: Action
7.2.7.1 Node
7.2.7.2 History State (h state)
7.2.7.3 The visualization of the Ground State (g state)
7.2.8 Cluster Elements
7.2.9 Interface
7.2.10 Controller Host Interface (CHI)
7.2.11 Message Passing and Shared Memory

7.2.12 Message Passing in Distributed Real-Time Systems
7.2.13 Interfaces of a Node Messages
7.2.14 Local Interfaces versus Linking Interfaces
7.2.14.1 Man-Machine Interface
7.2.14.2 Other Interfaces
7.2.15 Message Classification
7.2.16 Interface Specification
7.2.17 Elementary vs. Composite Interfaces
7.2.18 Information Push vs. Information Pull Interfaces
7.2.19 Time-Triggered Architecture
7.2.20 Real-Time (RT) Entities, Real-Time (RT) Images, and Real-Time (RT) Objects
7.2.21 Real-Time (RT) Entity, Attributes of Real-Time (RT) Entities
7.2.21.1 Real-Time (RT) Entity
7.2.21.2 Attributes of Real-Time Entities
7.2.21.3 Observation of a Real-Time Entity
7.2.21.4 Events Observations
7.2.21.5 Modelling State vs. Event Observation
7.2.21.6 Real-Time (RT) Images
7.2.21.7 Real-Time Object
7.2.21.8 Distributed RT Object

8. Real-Time Operating Systems

8.1 Real-Time Operating System Characteristics and Requirements
8.2 Process in Real-Time Systems
8.3 Concurrency in Embedded Systems
8.4 The Process States and Process Management
8.4.1 Kernel and Process Dispatching

8.5 **Principles of Real-Time Communication Networks**

8.5.1 Requirements of Communication Protocols
8.5.2 Systems Flow Control
8.5.3 Implicit Flow Control
8.5.4 Explicit Flow Control
8.5.5 Causes for Thrashing
8.5.6 Message Timing Models
8.5.6.1 Periodic Messages
8.5.6.2 Sporadic Messages
8.5.6.3 Aperiodic Message
8.5.6.4 Periodic and Non-Periodic Messages
8.5.7 Timely Blocking
8.5.8 Guarding Windows
8.5.9 Event-Triggered (ET) Protocols
8.5.10 Time-Triggered (TT) Protocols

Laboratory Description and Equipment

To cover the online format of the module design, remote labs are also designed, which are aligned with the pandemic situations. Bridge the gap from theory to practical implementation by performing a practical experiment in the lab. The practical part depends on the knowledge of practical tasks of the lecture contents (e.g., programming of embedded Systems with microcontrollers, scheduling, memory management, and time analysis). The Embedded Control Lab can be held on by the university of Siegen using an online platform.

Course References

[1]. Lee EA, Seshia SA. Introduction to embedded systems: A cyber-physical systems approach. MIT Press; 2017.

[2]. Kopetz H. Real-time systems: design principles for distributed embedded applications. Springer Science & Business Media; 2011 Apr 15.

Evaluation and Assessment Methods

Assessments take various forms, including Quizzes and Multiple-choice questions after each module. The course will be passed by 80% of the quizzes.

The tasks lead to the production of the intellectual output and the applied methodology.

The methodology in creating the SkoPS courses will be as follows: The leading partner is mainly responsible for preparing the materials and will distribute tasks to the other partners based on their experiences and skills. All other partners will review the prepared materials and provide feedback to improve the materials' quality. A systematic review process with all partners involved will further enhance the quality of the courses.

Right at the beginning of the project (directly after the kick-off meeting), a requirement analysis will be carried out by all project partners to precise the content of the courses which are outlined in this proposal. A second meeting will be conducted six months after the kick-off meeting as an internal course development Hackathon. Each partner will invite one or two external experts for this Hackathon to get their feedback for improving the quality of the course.