Standards & Technical Processes III:
Control, States, and Interfaces

Lecture 17

The Essential Idea and General Context
Activity flow can be controlled by inserting logic.

- The essential idea is to insert if-then-else logic[1] at the end or beginning of a flow, and associate actions with the outcome.
  - The control structure is not an independent model.
  - Other concepts of control in MBSE include configuration management, interface control, and data access control.[2]
- The literature on controlled activity flow includes basic types[3]
  - Sequencing, loop, branch, merge, synchronisation
  - Triggering, cancellation and completion, termination
  - Control standards occur in other standards e.g., SysML

[1] Or algorithms more generally, e.g. in physical systems (continuous) control.
[2] For example, refer to ISO/IEC/IEEE 24641:2023 (MBSSE).
[3] These are also called patterns. See Workflow PatternsThe Definitive Guide | Books Gateway | MIT Press

Discussion Points

Key Concepts

Using Rules for control of system flow and process

Three types of control structures

Process logic, State logic

States, events, transitions

Interfaces

Key Topics

- Rule Modelling
  - ***Control Structures***
  - Structured English
  - Decision Trees
  - Decision Tables
- State Modelling
  - States and Events
- Interface Definition

Loughborough University

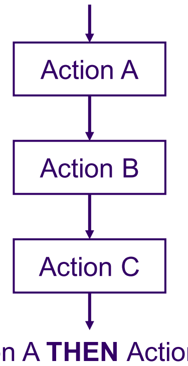
Rule Models for Control Logic

- Rules are used to specify the control of system flow.
  Control is accomplished by specifying conditions under which something is to be done i.e., for actions.
- Rules must be formulated as relationships amongst relevant system elements and must eventually be formulated as precise logics or algorithms.
- Imprecise formulation of rules corrupts the structure of control in the system.
- Rule models are usually not created as independent models but instead are embedded in the behaviour e.g.,
  - Activity model (e.g., Activity or Sequence Diagram)
  - Dynamics model (e.g., State Machine Diagram)

Loughborough University

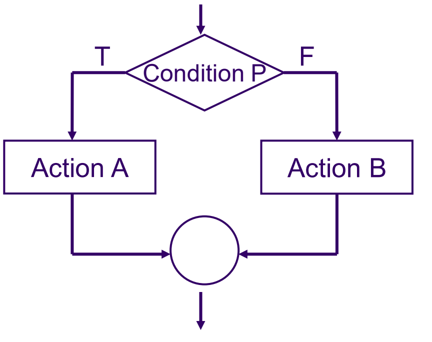
Control Structures for Sequences and Decisions

Sequencing: *a total order without branching or repetition*



Action A **THEN** Action B
THEN Action C

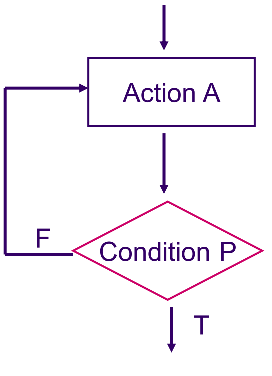
Decision: *selects one alternative according to the outcome of a decision*



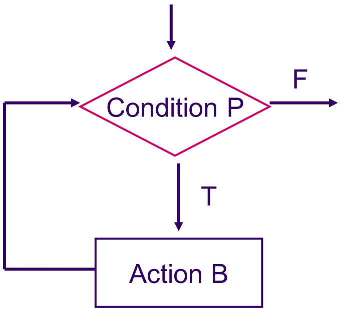
IF Condition P **THEN** Action A
ELSE Action B

Loughborough University

Loops in Control Structures



REPEAT Action A **UNTIL** Condition P



WHILE Condition P Action B

Loughborough University

Discussion Points

Key Concepts

Using Rules for control of system flow and process

Three types of control structures

Process logic, State logic

States, events, transitions

Interfaces

Key Topics

- Rule Modelling
  - Control Structures
  - ***Structured English***
  - Decision Trees
  - Decision Tables
- State Modelling
  - States and Events
- Interface Definition

Loughborough University

Structured English Approach

- Structured English is based on a theorem that guarantees that any *flow chartable* logic can be represented using only the <u>three</u> types of control structures (i.e. sequencing, decision, iteration).*
- Structured English expresses process logic by using the nested pattern

  If – then – else – so
- An action is expressed in structured English by an
  - imperative sentence or command or a
  - box (graphical form)

*C. Boehm and G. Jacopini, “Flow diagrams, Turing machines and Languages with two formation rules, “ Comm. of the ACM, vol 9, no. 5, May 1966, pp. 366.371

Loughborough University

Example of Rule Specification*
… when culinary art meets control logic!

Vegetables that are both leafy and crispy should be fried, while those that are crispy but not leafy should be boiled.

Prior to cooking, all vegetables that are not crispy should be chopped.

Then those that are green and hard should be boiled, while those that are hard but not green are steamed; those that are not hard are grilled.

STRUCTURED ENGLISH

- if crispy
  - if leafy then fry
  - otherwise boil
- otherwise chop
  - if hard
    - if green then boil
    - otherwise steam
  - otherwise grill

Note: "otherwise" has been substituted for "else".

*This quirky example is from legacy deliveries of the System Architecture module, prior to 2007. The reference was 'Solverg and Kung, Chapter 2', but this book is currently not available. The example however has been popular with students and carried forward.

Loughborough University

Discussion Points

Key Concepts

Using Rules for control of system flow and process

Three types of control structures

Process logic, State logic

States, events, transitions

Interfaces

Key Topics

- Rule Modelling
  - Control Structures
  - Structured English
  - ***Decision Trees***
  - Decision Tables
- State Modelling
  - States and Events
- Interface Definition

Loughborough University

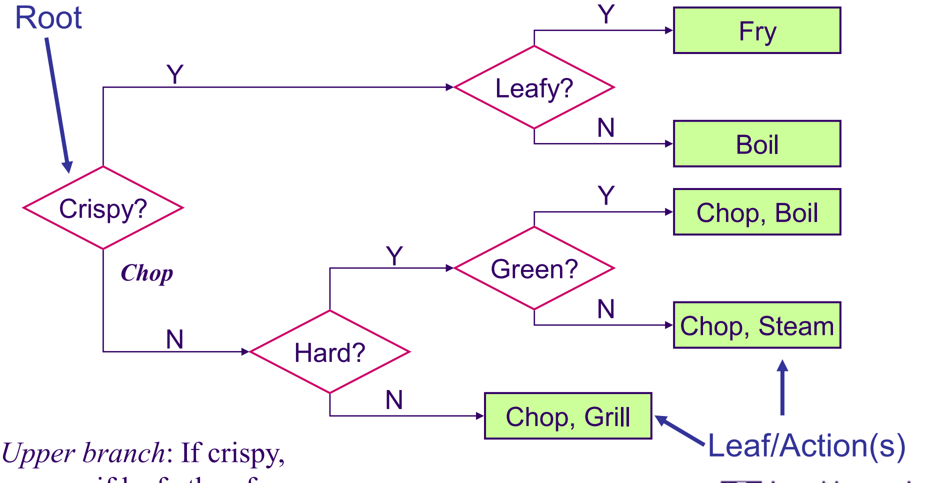
Decision Trees

- A decision tree represents a decision control structure in the form of a tree.
- A tree has a single root, branches and leaves
  - The root represents the first decision
  - Each node represents a decision point
  - The outcomes of a decision are shown as branches
  - The leaves of the tree represent the actions
- A path from the root to a leaf represents a *decision rule*.
- Each rule has a specified action or actions.*

**A single rule can result in multiple actions.*

Loughborough University

Decision Trees
(Can be implemented in Activity Diagrams*)



Upper branch: If crispy, if leafy then fry otherwise boil

**Refer to the Modelling Language lectures.*

Loughborough University

Discussion Points

Key Concepts

Using Rules for control of system flow and process

Three types of control structures

Process logic, State logic

States, events, transitions

Interfaces

Key Topics

- Rule Modelling
  - Control Structures
  - Structured English
  - Decision Trees
  - ***Decision Tables***
- State Modelling
  - States and Events
- Interface Definition

Loughborough University

Decision Tables

- Structured English can use all three types of control.
- Decision tables represent a *decision* control structure in tabular form and,
  - Represent the conjunctions and outcomes of the logic
  - Can be derived from decision trees
  - And like trees, assume the condition or decision variables take on a limited number of discrete values.
- Decision tables are
  - Well suited for specifying actions to take under complex combinations of conditions and are
  - Extensible and easier to visualise than complex trees

Loughborough University

The Four Parts of a Decision Table

lists the conditions or the decision variables

contains entries for the values associated with each condition or decision variable

	R_1 R_2 … R_N
Condition Stub	**Condition Matrix**
Action Stub	**Action Matrix**

lists the independent actions

links the actions to combinations of conditions shown in the condition matrix

Loughborough University

Rule Specification Example

Limited Entry Decision Table: Culinary Example

	R_1	R_2	R_3	R_4	R_5
Crispy	Y	Y	N	N	N
Hard	-	-	Y	Y	N
Leafy	Y	N	-	-	-
Green	-	-	Y	N	-
Fry	X				
Chop			X	X	X
Boil		X	X		
Steam				X	
Grill					X

Y : Yes - : Not relevant R_1, R_2,… rules (if… then…)

N : No X : Action Marker

Loughborough University

Discussion Points

Key Concepts

Using Rules for control of system flow and process

Three types of control structures

Process logic, State logic

States, events, transitions

Interfaces

Key Topics

- Rule Modelling
  - Control Structures
  - Structured English
  - Decision Trees
  - Decision Tables
- State Modelling
  - ***States and Events***
- Interface Definition

Loughborough University

States of a System

- Are associated with the *values* of the system attributes and activities (or operations)
- A simple light switch has *two states*: On and Off.
  - The state space consists of these two discrete states.
  - The time to switch between states is negligible.
- The autopilot on an automobile has *four states*:
  - Off
  - On
    - Accelerate
    - Cruise
    - Decelerate
- A chess board and its pieces have a *finite set of states*.
- In physics, particles can have an infinite number of states.

Loughborough University

Events

- An *event* is a primitive concept
- An event should be thought of as occurring instantaneously and causing *transitions* from one state to another
- An event may be identified with:
  - A specific *action* taken
  - A spontaneous occurrence dictated by nature
  - The result of a set of *conditions* being satisfied
- A discrete event system has a discrete set of events associated with it
- Examples: The model of the switch has only one event - the flipping of the switch. The events in the chess game are the set of allowable moves by the players

Loughborough University

Relation of States and Events

- A state diagram relates events and states
- A change of state caused by an event is called a transition
- A state diagram is a graph whose nodes are states and whose directed arcs are transitions labelled by event names
  - A state is drawn as a box, or a rounded box, or an oval (different authors)
  - The name of the state is inscribed in the node symbol
  - An Activity (of long duration) can be associated with a state; the activity starts when the system enters that state and ends when it exits

Loughborough University

Events And Actions

- A transition is drawn as an arrow from a state to the next state
  - The label on the arrow is the name of the event causing the transition
  - If multiple arrows leave a state, then each must correspond to a different event
  - Actions (instantaneous) are shown as labels adjacent to the event labels; the two are separated by a slash or a horizontal line
  - Rules may be shown inside brackets; sometimes they are called Guard Functions. The rule must evaluate as True for the transition to occur.

Loughborough University

Events And Actions (Cont’d)

- When an event occurs that is associated with a transition from a given state, the system will go to the state indicated by the transition arc and will simultaneously perform the indicated action.
- Transitions are sometimes self-loops: the arc starts and ends on the same state. This means that the event causes an action, but not a change of state.
- An event can also cause a change of state but not produce an action.

Loughborough University

Discussion Points

Key Concepts

Using Rules for control of system flow and process

Three types of control structures

Process logic, State logic

States, events, transitions

Interfaces

Key Topics

- Rule Modelling
  - Control Structures
  - Structured English
  - Decision Trees
  - Decision Tables
- State Modelling
  - States and Events
- ***Interface Definition***

Loughborough University

What is an interface?

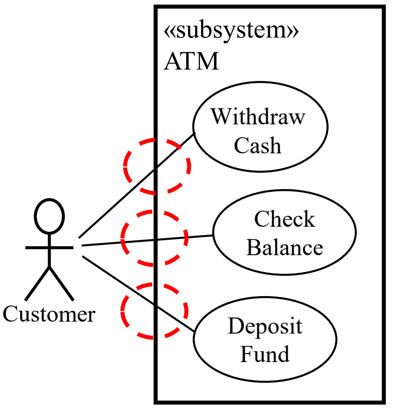
- '*The system boundary that is presented by a system for interaction with other systems.'**
- '*To connect two or more components for the purpose of passing information from one to the other***
- Recall the definition of System… ***interacting elements***…
- How is an interaction between elements achieved? – through an interface
- Hence, an interface is an enabler of interaction between
  - An element in the system and its environment – External Interface
  - Two elements of the system – Internal Interface
- An interface can be the natural (physical) boundary, or it can be treated a (sub-)system on its own right, e.g., user interface system in an aircraft cockpit.

* Fosse and Delp, 2013, Systems Engineering Interfaces: A Model Based Approach, IEEE
** ISO/IEC/IEEE. 2009 Systems and Software Engineering - System and Software Engineering Vocabulary

Loughborough University

Interface Identification

- Every association representing interaction between the system and the environment in the Use Case diagram indicates the potential need for a dedicated external interface.
- If UML diagrams are created consistently, the interactions are preserved through Use Case diagram to Activity diagram to Sequence diagram, but the details are refined.



Loughborough University

Interface Modelling

- Design of the interface then depends on the actual content being exchanged to achieve the intended interaction
  - Information
  - Energy
  - Material
- This should be made clear in the Sequence diagrams
  - What is exchanged?
  - What is the trigger?
  - What happens afterward?
- Specification of the final design is achieved through modelling in SysML Block Diagrams* using
  - Ports, and
  - Flow between the Ports
  - As well as the Requirement Diagram for documentation and traceability

**Refer to the Modelling Language lectures for further details.*

Loughborough University

Questions?