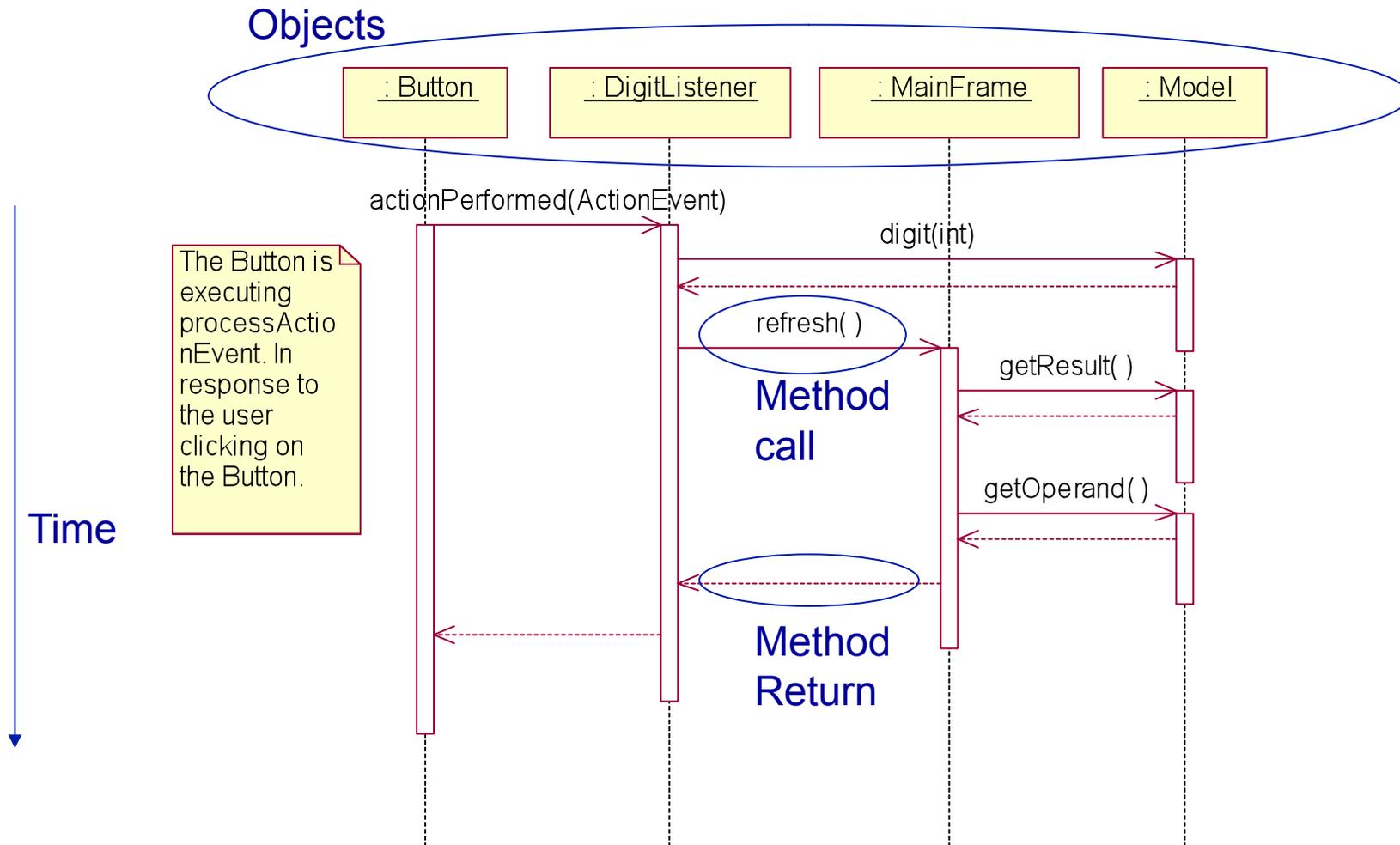# Sequence and Communication Diagrams

- Class Diagrams give a static view of the system. Time is not involved.

- We need also a way to describe the dynamics of the system.

- Sequence diagrams describe typical sequences of method calls.
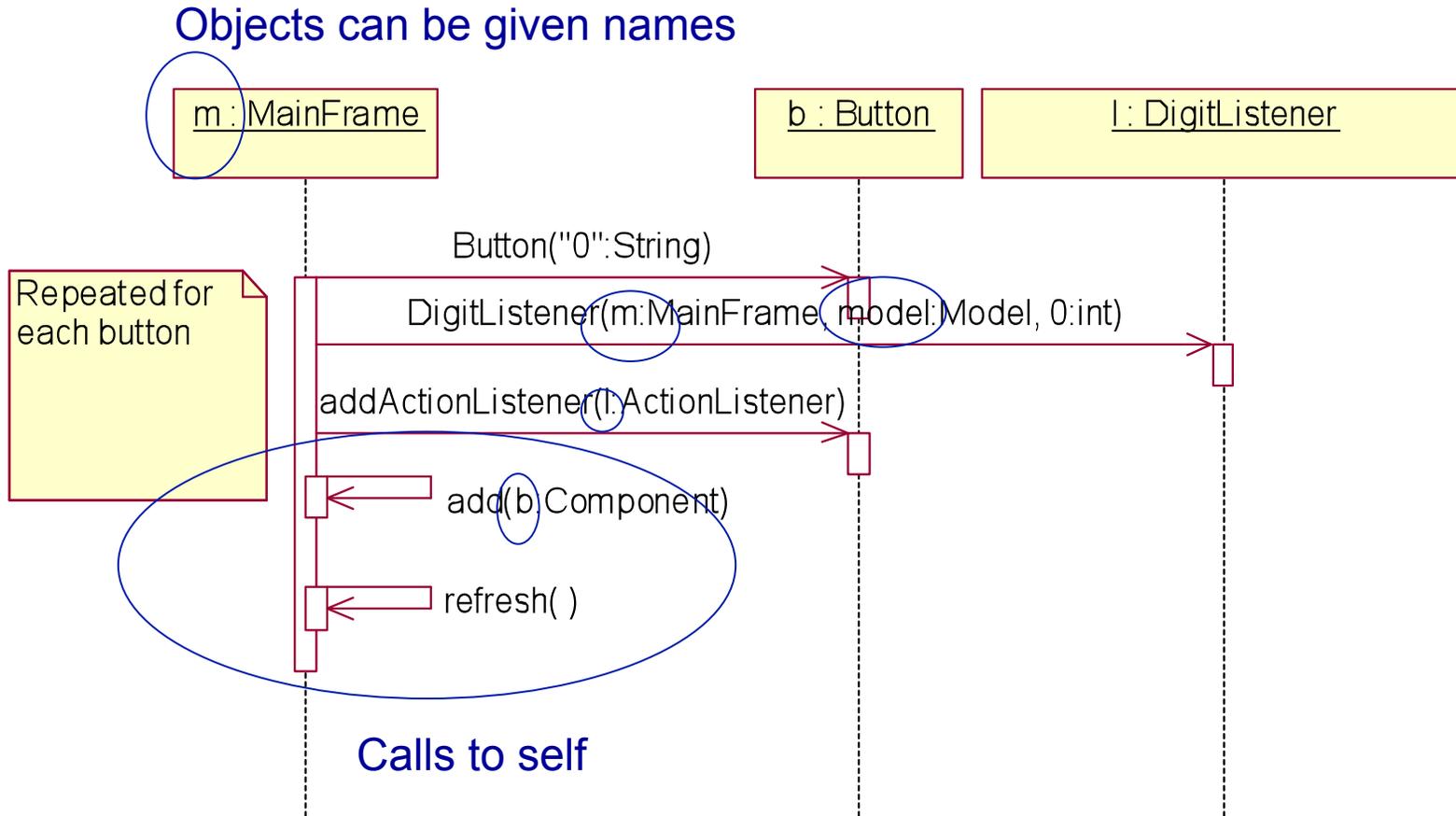
- Example

A sequence diagram

Objects

: Button   : DigitListener   : MainFrame   : Model

actionPerformed(ActionEvent)

digit(int)

The Button is executing processActionEvent. In response to the user clicking on the Button.

refresh( )

Method call

getResult( )

getOperand( )

Method Return

Time

Important point

- A sequence diagram shows an ordered tree (or forest) of invocations.

- Aside:
  - An "ordered tree" is a tree in which the order of the children matters.
  - An "ordered forest" is a sequence of ordered trees.

- Drawing tools let you make all kinds of crazy sequence diagrams. Be sure that yours is a proper tree (or forest)

Memorial University

Named object and Calls to self

Objects can be given names

m : MainFrame   b : Button   l : DigitListener

Repeated for
each button

Button("0":String)

DigitListener(m:MainFrame, model:Model, 0:int)

addActionListener(l:ActionListener)

add(b:Component)

refresh( )

Calls to self

Sequence Diagrams …

- Show a possible tree of calls.

- Do not (typically) show alternatives and looping

- Hint at but do not describe algorithms

- (For algorithms use pseudo code and/or activity charts and/or state diagrams)

- Are good at showing typical or important interactions between a few objects.

Sequence diagrams and abstraction

- Sequence diagrams cut across level of abstraction.
- Thus they should be used with care.

Communication Diagrams

- Show the same information as sequence diagrams and also links between the objects.

- Show the ordered tree structure of calls by hierarchical numbering.