Composite and Visitor Pattern

Jason
Computer structure

- Every part has value
- The value of a component part is the sum of all its (recursive) sub parts
Component

defered class
    COMPUTER_COMPONENT

feature -- Access

  value: INTEGER is
    -- Value of this computer part
    deferred
    ensure
      good_result: Result >= 0
  end

end
Composite

class
  COMPUTER_COMPOSITE
inherit
  COMPUTER_COMPONENT

feature -- Access
  parts: LINKED_LIST [COMPUTER_COMPONENT]
    -- Parts in Current composite

value: INTEGER is
  -- Value of this composite part
  do
    do
      from
        parts.start
    until
        parts.after
    loop
      Result := Result + parts.item.value
    parts.forth
  end
end

end
CPU

class
  CPU
inherit
  COMPUTER_COMPONENT
    redefine value end

feature -- Access

  value: INTEGER
    -- Value of this computer part

feature - Operations

  perform_addition is ...

  perform_substraction is ...

end
Monitor

class
    MONITOR

inherit
    COMPUTER_COMPONENT
    redefine value end

feature -- Access

    value: INTEGER
        -- Value of this computer part

feature - Operations

    display is ...

end
Assemble a computer

```python
graph_card: COMPUTER_COMPOSITE
main_memory, graph_memory: MEMORY

cpu: CPU
monitor: MONITOR
computer: COMPUTER_COMPOSITE
...

-- Initialize graph card.
graph_card.parts.extend (gpu)
graph_card.parts.extend (graph_memory)

-- Initialize computer.
computer.parts.extend (cpu)
computer.parts.extend (main_memory)
computer.parts.extend (monitor)
computer.parts.extend (graph_card)

-- Print value.
print (computer.value)
```

What if we just want value of memories?
What if we just want value of CPU or GPU?
What if ...
Visitor Pattern

- Process the elements of an (unbounded) data structure.
- Apply computations depending on the type of data node.
- Store the code outside of the data structure.
- Most of the time used together with the composite pattern.
deferred class
   COMPUTER_VISITOR

feature -- Process

   process_cpu (a_cpu: CPU) is deferred end
   -- Process a_cpu.

   process_monitor (a_monitor: MONITOR) is deferred end
   -- Process a_monitor.

   process_memory (a_memory: MEMORY) is deferred end
   -- Process a_memory.

   process_gpu (a_gpu: GPU) is deferred end
   -- Process a_gpu.
process_composite (a_composite: COMPUTER_COMPOSITE) is
   -- Process a_composite.
   do
      from
      a_composite.parts.start
      until
      a_composite.parts.after
      loop
         a_composite.parts.item.process (Current)
         a_composite.parts.forth
      end
   end
end
Component

defered class
  COMPUTER_COMPONENT

feature -- Process

  process (a_visitor: COMPUTER_VISITOR) is
    -- Process Current using a_visitor.
    deferred
    end
  end

end
CPU

class CPU
inherit COMPUTER_COMPONENT

feature -- Access

value: INTEGER
-- Value of this computer part

feature - Operations
perform_addition is ...
perform_substraction is ...

feature -- Process

process (a_visitor: COMPUTER_VISITOR) is
-- Process Current using a_visitor.
do
   a_visitor.process_cpu (Current)
end

end
Composite

class COMPUTER_COMPOSITE
inherit COMPUTER_COMPONENT

feature -- Access

parts: LINKED_LIST [COMPUTER_COMPONENT]
  -- Parts in Current composite

feature -- Process

process (a_visitor: COMPUTER_VISITOR) is
  -- Process Current using a_visitor:
    do
      a_visitor.process_composite (Current)
    end

end
class MEMORY_VALUE_VISITOR
    inherit COMPUTER_VISITOR

feature -- Calculate

memory_value (a_component: COMPUTER_COMPONENT): INTEGER is
    -- Value of all kinds of memory in a_component
    do
        last_value := 0
        a_component.process (Current)
    end

feature -- Access

last_value: INTEGER
    -- Last calculated value
feature -- Process

process_memory (a_memory: MEMORY) is
  do
    last_value := last_value + a_memory.value
  end

process_cpu (a_cpu: CPU) is
  do
  end

process_monitor (a_monitor: MONITOR) is
  do
  end

process_gpu (a_gpu: GPU) is
  do
  end

end