Learning Design Execution of Complex Learning Processes

Juan Manuel Dodero
DEI Research Group
Universidad Carlos III de Madrid

[DP plenary meeting, OTEC, OUNL, May 2006]
Context, Objectives and Scope

- **Context**
  - Specification of Learning Processes
  - EML and LD

- **Objectives**
  - Review LD languages wrt. LP execution issues
  - Present a language used to describe the composition and execution of complex learning processes

- **Scope of this presentation**
  - Machine-understandable
  - Learning Design and Execution
Agenda

1. **Characterize** ‘complexity’ of LP descriptions
   - Restrict to one complexity dimension
   - Describe some execution issues

2. **Review** execution issues of LD languages
   - Complex activity structuring
   - Learning flow control
   - Synchronization

3. **Present** our approach: LPCEL
   - Project-oriented Learning case study

4. **Conclusions**
Complex Learning Process (CLP)

- **Concept of CLP**
  - **Unanticipated** integration of **mixed** learning resources and methods, based upon the **collaboration** of instructors and learners, within a **large** learning process

- **Complexity depends on...**
  - **Largeness** = how complex are the learning activities per se?
    - Long run
    - Many activities, participants, and relationships between
  - **Collaboration** degree
    - Autonomy of learner?
    - Who designs and controls the learning process?
  - **Mixing** degree
    - Multi-disciplinary
    - Varied pedagogical methods
  - **Anticipation** of design
    - Learning design-time vs. learning-time
Complexity of a LP description

- Ariadne CDF
- IFIS LMML
- UNED PALO
- JISC TML
- OUNL EML [IMS LD]
- [Part of] SCORM
- LAMS

Design and control of the process

- Selection of activities and outcomes
- Planning of activities

Complexity of the Task

- One discipline
  - Few days/weeks
  - One didactic method
- Multidisciplinary
  - Long-run learning
  - Several didactic methods

Autonomy of the Learner

- Selection of topic
- Definition of learning results

Instructor
Learner
Some complexity features

- Activity structures
  - Complex?
- Diverse learning flows
  - Concurrent?
- Design-time descriptions
  - Run-time execution?
  - Can be changed/composed? During run-time?
Execution issues: map

Complex structuring

Complex activities
Parallel activities
Branching
Looping
Transactions
Synch points
Split runs

Learning flow
Synchronization

Clue:

Requirement/situation to model
Rule from LD specifications
Description of issues

Examples from a Project-Oriented Learning case study
Issue 1: Complex activities

Diverse kinds of complex activities: sequence, selection, parallel branching, split activities, etc.
“During the transition phase, the project group must beta-test the product, write user and administration manuals, and deploy the final system”

IMS LD activity-structures
- Activities can be sequenced/selected

LAMS LD complex-activities
- Diverse kinds: sequence, parallel, optional, etc.

Types of activities
- Sequence and selection: ok
- Parallel branching: only at play or role-part level
- What about other/future activity types? (e.g. assessment, looping, split, synchronized, etc.)
Issues 2+3: Flow control

Branching + Looping
“A typical project will go through several iterations. The Engineering workflow is organized in Requirements, Analysis, etc. The project manager manages the iterations”

IMS LD Level B
• Branching on learning flows: properties + conditions
• Once the user indicates the activity to be completed, then this activity stays completed in the run

Flow control primitives’ abstraction level
• Low-level flow control primitives
Iterated activities cannot be split
• Workaround: designer-managed ‘completion status’
• Flow control criteria are not always simple
Issues 2+3: Flow control

Non-simple flow control

“During the construction phase, the product is checked against the quality level set in the inception phase. If it does not meet this level, the entire cycle in the construction phase begins again”
Issue 4: Parallel activities

Parallel activity running
“During the transition phase, the project group must [simultaneously] beta-test the product, write user and administration manuals, and deploy the final system”

IMS LD role-parts
- Each role-part associates exactly one role with exactly one type of activity
- The same role may only be referenced once in the same act
- If multiple activities [...] need to be associated for the same role an activity-structure or wrapper environment should be used

LAMS LD group branching

Granularity of ‘containers’ of parallel activities
- A role-part for each activity/activity structure. All parallel role-parts in the same act.
Parallel activities in IMS LD

<table>
<thead>
<tr>
<th>Consultant</th>
<th>Project Manager</th>
<th>Documentarist</th>
<th>Sys Admin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Act 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Role-part</td>
<td>Beta testing</td>
<td>Quality testing</td>
<td></td>
</tr>
<tr>
<td>Act 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Role-part</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Act 3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Role-part</td>
<td>Write manuals</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Develop user manual
Develop Admin manual
Deployment
**Issue 5: Activity synchronization points**

**Synch point between plays/acts/activities**
“Develop introductory lessons concurrently with the project-oriented learning process, so things are gradually learnt as they are applied during each project iteration”

**IMS LD synchronization**
- Plays always run concurrently. Acts always run in sequence
- Synch can be on-completion (user choice/time limit) or when-a-property-is-set

**LAMS LD gates**
- Transitions between activities can be set with *gates*, which are released on *permissions* (wait for staff), *schedule* (set time), or *synchronization* (all students reach the point)

**Explicit synch points**
- Implicit synch between acts. Explicit synch (e.g. milestones) between activities?
- LAMS Gates: only three kinds of activity synchronization conditions
Issue 5: Activity synchronization points

**Activity-based synchronized ‘rendezvous’**

“During the construction phase, the product is checked against the quality level set in the inception phase. If it does not meet this level, the entire cycle in the construction phase begins again”
**Issue 5: Activity synchronization points**

**Guard-based synchronization milestone**

“The [so-called] ‘architecture milestone’ is reached after Elaboration activities [project planning, problem domain analysis, software architecture, use case modeling] are completed, and use cases are 80% described”

<table>
<thead>
<tr>
<th>Project Manager</th>
<th>Analyst</th>
<th>Architect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture milestone</td>
<td>80% complete</td>
<td></td>
</tr>
</tbody>
</table>

[Diagram showing the relationship between Project Manager, Analyst, and Architect with activities and milestones.]

- Elaboration
  - Project planning
  - Problem domain analysis
  - Use case modelling
  - Software architecture
Issue 5: Activity synchronization points

**Activity running-based conditional synchronization**

“Start the lesson *before* the first project iteration [...]. Start the next project iteration *after* the external release of the previous iteration”
Issue 6: Split-running parallel activities

Continuous activities (e.g. management and control)
“The project manager role manages the phase transition and progress of the project”

LD notifications
- IMS LD Level C notifications sent can reference to (already activated) activities

Synchronized continuous activities
- Some activities have not well-defined start and end times, but take the control focus only in some intervals.
- Activity notifications activate the same activity run again.
Issue 7: Learning transactions

**Commit and rollback learning transactions**
“The learner must (1) study and pass a test on theoretical concepts and (2) realize and pass a set of collaborative practical assignments in order to complete the Subject”

**Transactions**
- Transactions support speculative unsynchronized computation on a copy of the data followed by a commit or abort

**Transaction primitives**
- IMS LD: synchronize with plays + acts
- LAMS LD: synchronize with gates
- Deadlocks? Begin/Commit/Rollback semantics
- Compensating transactions?
Learning transaction
Compensating transaction
Our approach: LPCEL

- LPCEL = Learning Process Composition and Execution Language
- Conceptual model
  - Central elements
  - Examples
- Framework/architecture
- How it is implemented
  - Namespaces
  - Software patterns
LPCEL execution elements

- Complex activity structures
- Learning flow control
- Parallel activities
- Synchronization points
- Conditional synchronization (guard)
- Split-running activities
- Learning transactions
Project-oriented Learning case study

**Evaluation**
- Certification
- Conduction

**Products**
- Exercise Topic 1
- Practical exercises Topic 2
- Definition of Work Structure
- Investigation Report Topic 2
- Practical exercises Topic 2
- Definition of Software Engineering Process
- Offer and Contract of Project
- Software Requirement Specification
- Analysis Model

**Individual Work**
- Presentation of the Course
- Definition of Project Plan
- Interview with clients of the project
- Exposition: Topic 1
- Exposition: Topic 2
- Exposition: Topic 3
- Exposition: Topic 4
- Exposition: Topic 5
- Exposition: Topic N

**Group Work**
- Definition of the Course
- Definition of restrictions of project
- Analysis Model
- WorkFlow Analysis

**Instructor**
- Certification WFR
- Consultation WFR
- Consultation UC
- Consultation MUC

**Consultant**
- Certification EV 1
- Conduction EV 1
- Certification EV N
- Conduction EV N
- Exposition: Topic 2
- Investigate Report: Topic 2
- Investigation Report: Topic 2
Unanticipated design of a CLP

Complex Learning Process (\(S_1\))

- Find Actors
- Find Use Cases
- Developed a UC Model
- Interview with clients of the project
- Definition of Project Plan
- Definition of Work Method
- Workflow:
  - Requirements
  - Analysis
  - Start
  - Presentation of the Course
  - Definition of restrictions of project
  - Exposition: Topic 3
  - Exposition: Topic 4
  - Consultation UC
  - Consultation MUC
  - Consultant Instructor Group Work
  - Individual Work
  - Exposition: Topic N
  - Practical exercises Topic 1

Complex Learning Process (\(S_2\))

- Find Use Cases
- Developed a UC Model
- Interview with clients of the project
- Definition of Work Method
- Workflow:
  - Requirements
  - Analysis
  - Start
  - Definition of restrictions of project
  - Exposition: Topic 4
  - Consultation UC
  - Consultation MUC
  - Consultant Instructor Group Work
  - Individual Work
  - Exposition: Topic N
  - Practical exercises Topic 1

Complex Learning Process (\(S_n\))

- Find Use Cases
- Developed a UC Model
- Interview with clients of the project
- Definition of Work Method
- Workflow:
  - Requirements
  - Analysis
  - Start
  - Definition of restrictions of project
  - Exposition: Topic 1
Example: Flow control

**Conditional + iterated activities**

```xml
<lpcel:assessment-activity identifier="AA-Quality-Testing">
  <imsld:title>Quality assessment</imsld:title>
</lpcel:assessment-activity>

<imsld:activity-structure identifier="AS-Engineering-workflow" structure-type="complex-activity">
  <imsld:title>Engineering iteration</imsld:title>
  <lpcel:do-while>
    <imsld:conditions>
      <imsld:complete>
        <lpcel:component-activity-ref ref="AA-Quality-Testing"/>
      </imsld:complete>
    </imsld:conditions>
    <lpcel:sequence>
      <lpcel:component-activity-ref ref="A-Requirements"/>
      <lpcel:component-activity ref="A-Analysis-Design"/>
      <lpcel:component-activity ref="A-Implementation"/>
    </lpcel:sequence>
  </lpcel:do-while>
</imsld:activity-structure>
```
Example: Synchronization

Guarded synch milestone in a project (i)

```xml
<lpcel:assessment-activity identifier="AA-Eval-Elaboration-Criteria">
    <imsld:title>Evaluate Elaboration Criteria</imsld:title>
    <lpcel:conditional-synch>
        <lpcel:guard>
            <imsld:is>
                <imsld:greater-than>
                    <imsld:property-ref ref="LR-Use-Case-Modelling"/>
                    <imsld:property-value>80</imsld:property-value>
                </imsld:greater-than>
            </imsld:is>
        </lpcel:guard>
        <imsld:complete>
            <lpcel:component-activity-ref ref="LA-Project-Planning"/>
            <lpcel:component-activity-ref ref="LA-Domain-Analysis"/>
            ...
        </imsld:complete>
    </lpcel:conditional-synch>
</lpcel:assessment-activity>
```
Example: Synchronization

Guarded synch milestone in a project (ii)

...  
<imsld:activity-structure identifier="AS-Elaboration-phase"  
                     structure-type="complex-activity">  
  <imsld:title>Elaboration phase</imsld:title>  
  <lpcel:sequence>  
    <lpcel:parallel>  
      <lpcel:component-activity-ref ref="A-Project-Planning"/>  
      <lpcel:component-activity-ref ref="A-Problem-Domain-Analysis"/>  
      <lpcel:component-activity-ref ref="A-Use-Case-Modelling"/>  
      <lpcel:component-activity-ref ref="A-Software-Architecture"/>  
    </lpcel:parallel>  
    <lpcel:assessment-activity-ref ref="AA-Eval-Elaboration-Criteria"/>  
  </lpcel:sequence>  
</imsld:activity-structure>
Example: Learning transaction

Commit/rollback/compensate learning transactions (i)

```xml
<lpcel:assessment-activity identifier="AA-Theory">
  <lpcel:sequence>
    <lpcel:component-activity-ref ref="LA-Study"/>
    <lpcel:component-activity-ref ref="LA-Concept-Test"/>
  </lpcel:sequence>
</lpcel:assessment-activity>

... <!--Similar for activity AA-Practice-->
...

<imsld:activity-structure identifier="AS-Subject"
  structure-type="complex-activity">
  <imsld:title>Elaboration phase</imsld:title>
  <lpcel:transaction-begin id="T-AS-Subject">
    <lpcel:parallel>
      <lpcel:component-activity-ref ref="AA-Theory"/>
      <lpcel:component-activity-ref ref="AA-Practice"/>
    </lpcel:parallel>
  </lpcel:transaction-begin>
</imsld:activity-structure>
```

(see next page...)
Example: Learning transaction

Commit/rollback/compensate learning transactions (ii)

(from previous page...)

<imsld:conditions>
  <imsld:if>
    <!--pass(AA-Concept-Test) and pass(AA-Practical-Test)-->
  </ imsld:if>
  <imsld:then>
    <lpcel:transaction-commit id="T-AS-Subject">
  </imsld:then>
  <imsld:else>
    <lpcel:transaction-rollback id="T-AS-Subject">
  </imsld:else>
</imsld:conditions>
<lpcel:transaction-end id="T-AS-Subject">
</imsld:activity-structure>
Project-Oriented Learning Case Study

Unanticipated Design of a CLP
Conclusions

• Objectives
  - Characterized CLP
  - Reviewed major LD execution issues
  - How LPCEL can be used to describe execution

• Contributions to LPCEL from Computer Science
  - Process workflow management
  - Concurrency control
  - Soft transactions
  - Software design patterns

• Other related work
  - Composition
  - Unanticipation