



***Automating Production of Cross Media  
Content for Multichannel Distribution***

**AXMEDIS Tool Core for  
MPEG-21 Authoring/Playing**

**1st AXMEDIS Conference**  
**Convitto della Calza, Florence, Italy**  
**30nov-2dec 2005**

1st AXMEDIS Conference, 30 Nov 2005, AXMEDIS Tool Core 1




### Summary

- **AXMEDIS Authoring Tool/Player Scenario**
- **MPEG21 Overview**
  - ♣ Digital Item (DIDL)
  - ♣ Rights Expression Language (REL)
  - ♣ Intellectual Property Management and Protection (IPMP)
- **AXMEDIS Tool Core**
  - ♣ General Architecture
  - ♣ Object Oriented Design
    - The Controller
    - The Protection Processor
  - ♣ Authoring Tool User Interface
    - Browsing the MPEG21 Digital Item
    - Editing the attributes of MPEG21 Digital Item elements
- **Conclusions**

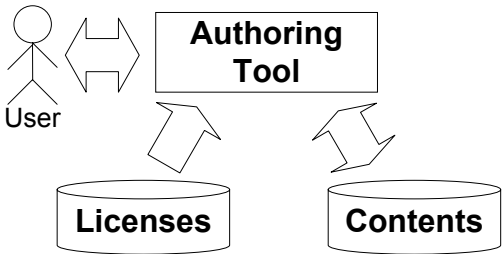
1st AXMEDIS Conference, 30 Nov 2005, AXMEDIS Tool Core 2

### Authoring tool scenario



**AXMEDIS**


- **A typical scenario for authoring tool**
  - ♣ the creation of a digital item
    - from other digital items
  - ♣ on the basis of acquired licenses (DRM)
  - ♣ embedding digital resources: e.g. mp3 files, pdf files, etc.
- **Authoring Tool is also a Player**
  - ♣ To be able to preview the authored content



```
graph TD; User((User)) <--> AT[Authoring Tool]; AT <--> Licenses[(Licenses)]; AT <--> Contents[(Contents)];
```

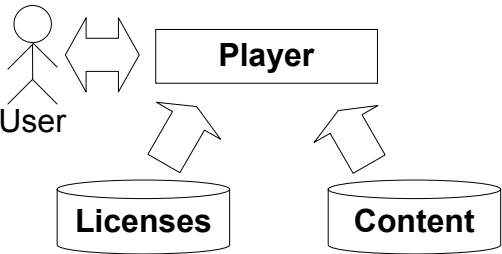
1st AXMEDIS Conference, 30 Nov 2005, AXMEDIS Tool Core 3

### Player scenario



**AXMEDIS**

- **A typical scenario for AXMEDIS player**
  - ♣ the rendering of a digital item
    - requires synchronization with other digital items content
  - ♣ on the basis of acquired licenses (DRM)
  - ♣ rendering embedded digital resources
    - e.g., mp3 files, pdf files, etc.



```
graph TD; User((User)) <--> Player[Player]; Player <--> Licenses[(Licenses)]; Player <--> Content[(Content)];
```

1st AXMEDIS Conference, 30 Nov 2005, AXMEDIS Tool Core 4

## MPEG21 Overview: Digital Item Concept

- Extremely flexible content packaging

The diagram illustrates the Digital Item Concept. A large grey rounded rectangle labeled "Container" contains three smaller rounded rectangles labeled "Item". Each "Item" contains a yellow box labeled "Descriptor". The first and second "Item" each contain two purple boxes labeled "Component". Each "Component" contains a yellow box labeled "Descriptor" and an orange box labeled "Resource". The third "Item" contains one purple "Component" box (with "Descriptor" and "Resource") and one yellow "Descriptor" box.

1st AXMEDIS Conference, 30 Nov 2005, AXMEDIS Tool Core

## MPEG21 Rights Expression Language

The diagram shows a central grey box labeled "right". Three arrows point from "right" to three other grey boxes: "principal" (labeled "issued to"), "resource" (labeled "associated with"), and "condition" (labeled "subject to").

- REL**
  - can declare rights and permissions
- REL grant consist of**
  - principal to whom grant is issued
  - rights the grant specifies
  - resource to which right in grant applies
  - condition to be met before grant can be exercised

1st AXMEDIS Conference, 30 Nov 2005, AXMEDIS Tool Core

## MPEG21 Intellectual Property Management and Protection

AXMEDIS

- Terminal can get IPMP Tools
  - embedded in content
  - by downloading it

**Digital Item**

**Digital Item Declaration**

**IPMP\_Control\_Info\_Descriptor**

IPMP Tool List

IPMP Tool ID(s)

Alternate IPMP Tool ID(s)

Parametric Tool Description(s)

IPMP Tool Holder

IPMP ToolBody

Rights Holder

Rights Tool ID(s)

Rights Expressions

IPMP\_Descriptor

⋮

IPMP\_Descriptor

Resource (s)

1st AXMEDIS Conference, 30 Nov 2005, AXMEDIS Tool Core

7

## AXMEDIS Tool Core: General Architecture


AXMEDIS

- **Model-View-Controller design pattern**
  - ♣ use the Observer (event driven) and Command design pattern
- **Model Status Manager**
  - ♣ ensure a correct manipulation of the Model
    - avoiding simultaneous read/write access to the same data (locks)
- **Controller**
  - ♣ intermediary of every action performed on the Model.
  - ♣ Executes/Inhibits operations with respect to rights and produce logs

1st AXMEDIS Conference, 30 Nov 2005, AXMEDIS Tool Core

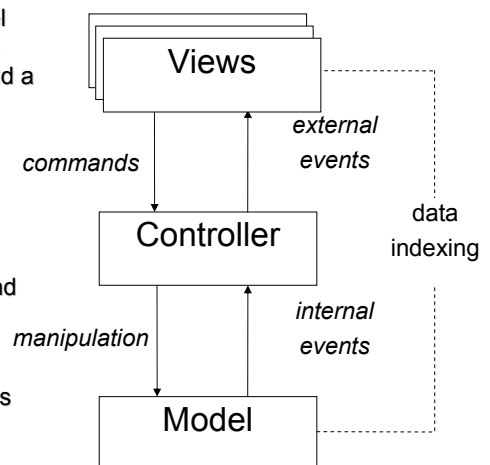
8

## AXMEDIS Tool Core: General Architecture (2)



AXMEDIS

- **Model data-indexing**
  - ♣ mechanism to refer the content outside the model
  - ♣ views can refer a specific element and does not hold a memory pointer to it
  - ♣ It allows only authorized access
- **Content manipulation walk-through**
  1. User acts on a view
  2. View prepares a command to be executed (using indexes)
  3. Controller checks authorization and resolves indexes and uses Model
  4. Model raises “change” events
  5. View updates itself




The diagram shows the MVC (Model-View-Controller) architecture. At the top is a stack of boxes labeled 'Views'. Below it is a box labeled 'Controller', and at the bottom is a box labeled 'Model'. Arrows indicate the flow of information: 'external events' point from the Views to the Controller; 'commands' point from the Controller to the Views; 'manipulation' points from the Controller to the Model; 'internal events' point from the Model to the Controller. Dashed lines represent 'data indexing' from the Model to the Views and from the Controller to the Model.

1st AXMEDIS Conference, 30 Nov 2005, AXMEDIS Tool Core

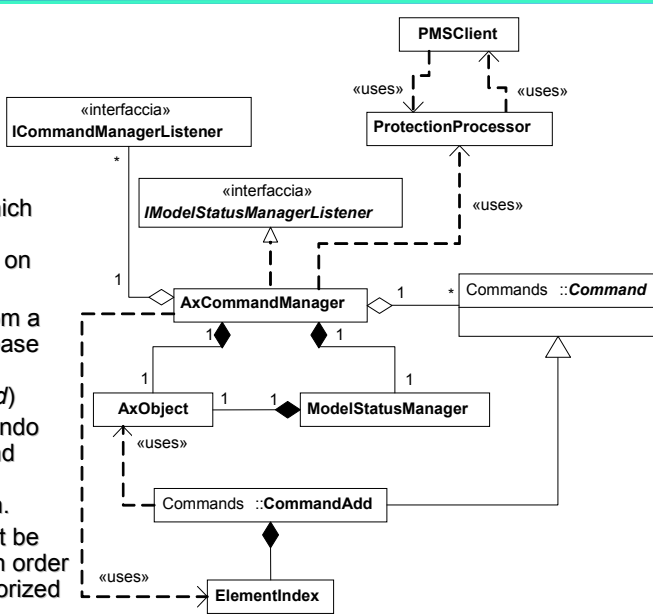
9

## AXMEDIS Tool Core: Controller



AXMEDIS

- **Commands**
  - ♣ actions which can be performed on the Model.
  - ♣ derives from a common base class (*Command*)
  - ♣ provides undo support and grant verification.
  - ♣ rights must be declared in order to be authorized



The UML class diagram illustrates the Controller's internal structure. It features several classes and interfaces:
 

- Interfaces:** `«interfaccia» ICommandManagerListener` and `«interfaccia» IModelStatusManagerListener`.
- Classes:** `PMSCClient`, `ProtectionProcessor`, `AxCommandManager`, `Commands ::Command`, `AxObject`, `ModelStatusManager`, `Commands ::CommandAdd`, and `ElementIndex`.
- Relationships:**
  - `PMSCClient` uses `ProtectionProcessor` (indicated by dashed arrows with «USES»).
  - `ProtectionProcessor` uses `AxCommandManager` (indicated by a dashed arrow with «USES»).
  - `AxCommandManager` implements `ICommandManagerListener` and `IModelStatusManagerListener` (indicated by solid arrows with open heads).
  - `AxCommandManager` has aggregation relationships (open diamonds) with `AxObject` and `ModelStatusManager`.
  - `AxObject` has aggregation relationships with `ModelStatusManager` and `Commands ::CommandAdd`.
  - `ModelStatusManager` has an aggregation relationship with `Commands ::CommandAdd`.
  - `Commands ::CommandAdd` inherits from `Commands ::Command` (indicated by a solid arrow with an open head).
  - `ElementIndex` uses `Commands ::CommandAdd` (indicated by a dashed arrow with «USES»).

1st AXMEDIS Conference, 30 Nov 2005, AXMEDIS Tool Core

10

## AXMEDIS Tool Core: Controller (2)

- **AddCommand contains two parameters,**
  - ♣ new element to be attached to the structure
  - ♣ the index of the parent element the new one should be added to
- **A view deals with ElementIndex objects**
- **It asks AxCommandManager to obtain data objects referred by indexes**
- **It receives a clone of the data element**
  - ♣ purified from references to the Model
- **It uses cloned data objects for rendering**

1st AXMEDIS Conference, 30 Nov 2005, AXMEDIS Tool Core

11

## AXMEDIS Tool Core: Protection Processor

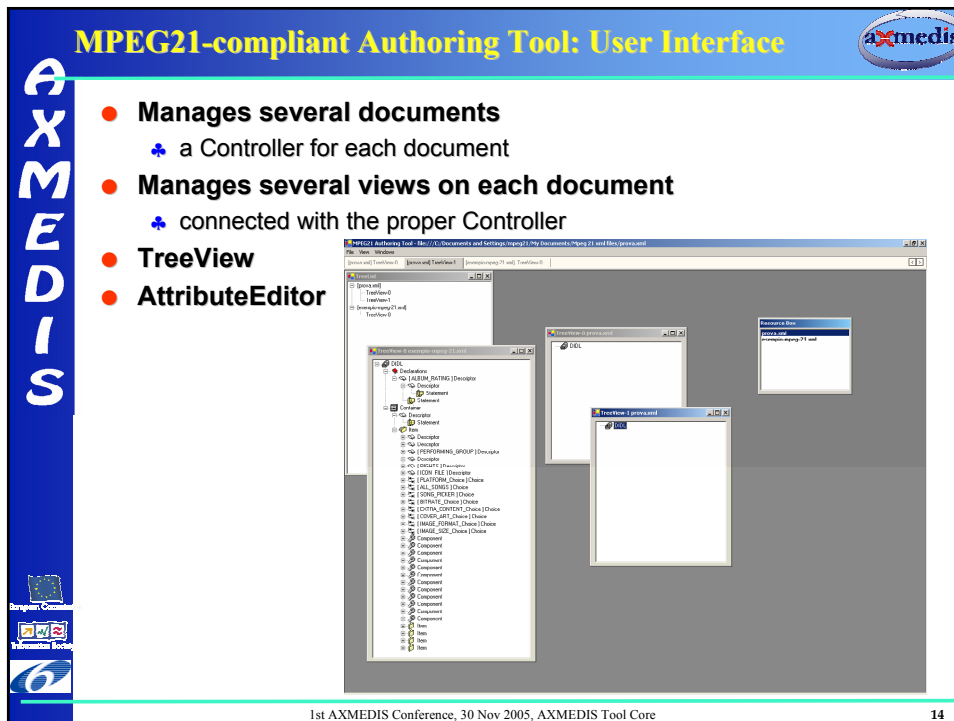
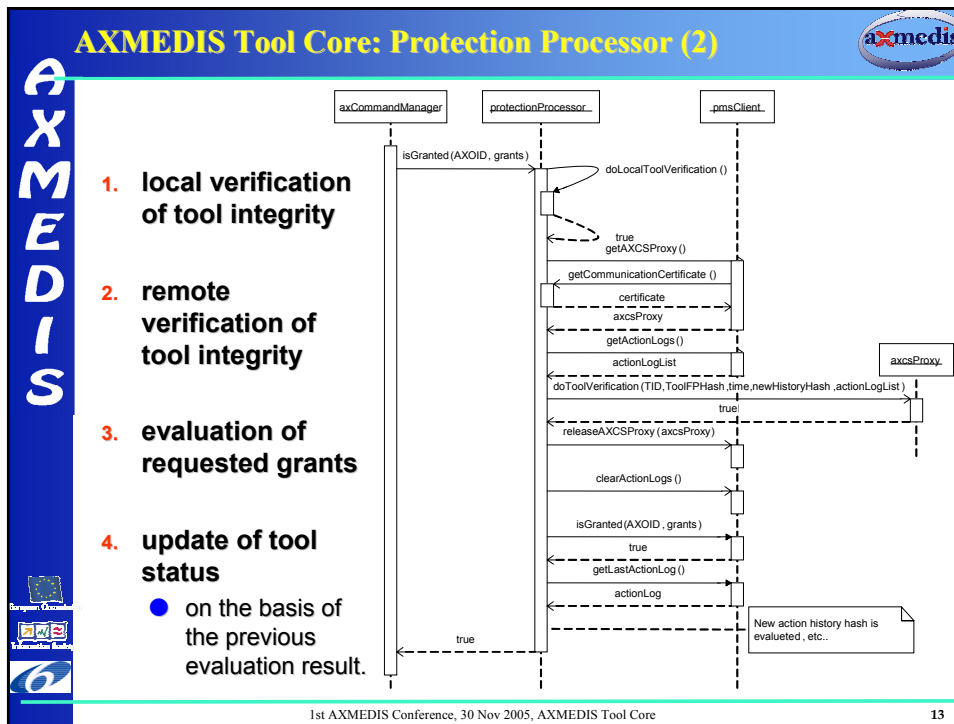
- **Protection Processor**
  - ♣ verifies the integrity of the software
  - ♣ protects/unprotects the elements of the Model.
  - ♣ MPEG21 IPMP
- **Protection Manager Support**
  - ♣ is a remote service
  - ♣ checks the grants needed to execute an action (authorization algorithm)
  - ♣ is used by Protection Processor through a built-in client
  - ♣ MPEG21 REL
- **DRM governed manipulation also in offline context**

```

classDiagram
    class AxCommandManager
    class ProtectionProcessor {
        +unprotectElement(in protectedElement : AxElement) : AxElement
        +protectElement(in unprotectedElement : AxElement) : AxElement
        -ProtectionProcessor()
        +getInstance() : ProtectionProcessor*
        +isGranted(in AXOID, in grants) : bool
        +doToolVerification() : bool
        -doLocalToolVerification() : bool
        +verifySoftware(in swld : char*) : int
        -doToolCertification() : void
        +addActionLog(in actionLog : char*) : void
        +getUID() : char*
        +getTID() : char*
        +getToolFingerprint() : char*
        +getSystemTime() : char*
        +disableTool() : void
        +getCommunicationCertificate() : char*
    }
    class ProtectionManagerSupportService
    AxCommandManager --> ProtectionProcessor : <<uses>>
    ProtectionProcessor --> ProtectionManagerSupportService : <<uses>>
    
```

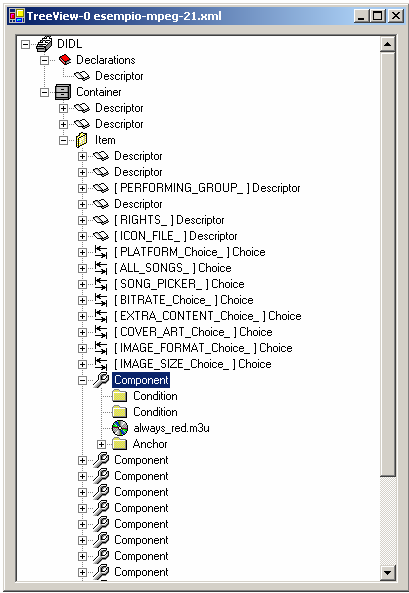
1st AXMEDIS Conference, 30 Nov 2005, AXMEDIS Tool Core

12



### Browsing an MPEG21 Digital Item

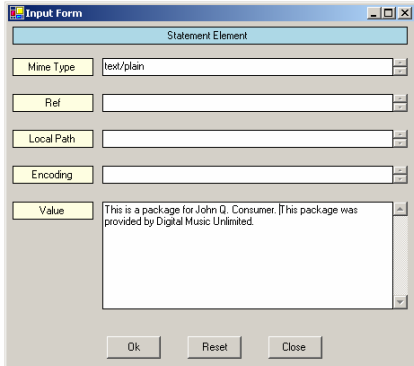
- renders the hierarchical structure of MPEG-21 DI
- adds/deletes DI elements
- cut-copy-paste and drag-drop
- exploits Commands provided by the Controller
  - Expand, Add, Delete, Move, Copy...



1st AXMEDIS Conference, 30 Nov 2005, AXMEDIS Tool Core



### Editing attributes of the MPEG21 Digital Item elements

- dynamically configured on the basis of the target element
  - the schema of the MPEG21 element
- exploits Commands provided by the Controller
  - Edit






1st AXMEDIS Conference, 30 Nov 2005, AXMEDIS Tool Core





## Conclusions

- The MPEG21 standard has been intensively studied
- The tool core can manipulate MPEG21 digital items following the standard guidelines
- The architecture benefits of the most advanced design patterns
- The Authoring Tool has been easily developed on the basis of the Command Manager interface
- A flexible approach to DRM has been realized
  - ♣ Remote service or local verification
  - ♣ It considers device/application trustiness
- The tool core architecture has been used to realize the AXMEDIS Editor



1st AXMEDIS Conference, 30 Nov 2005, AXMEDIS Tool Core 17