The 17th International Conference on 3D Web Technology
WEB 3D 2012

Virtual Reality Procedure Training Simulators in X3D

Olavo Belloc, Rodrigo Ferraz, **Marcio Cabral**, Roseli Lopes, Marcelo Zuffo

CAVERNA *Digital*

Escola Politécnica

Universidade de São Paulo

Agenda

- Introduction
- Related Work
- Proposed Nodes
- Results and Live Demo
- Conclusions

Introduction

- Virtual Reality training applications have become increasingly common – [Blümel et al. 2005][Lin et al. 2002]
- Emphasis:
  - on training individuals for complex or risky procedures
  - on training individuals' dexterity
- **Procedures** – Involves a complex set of actions that must be performed in a specific order or under very specific conditions

Introduction

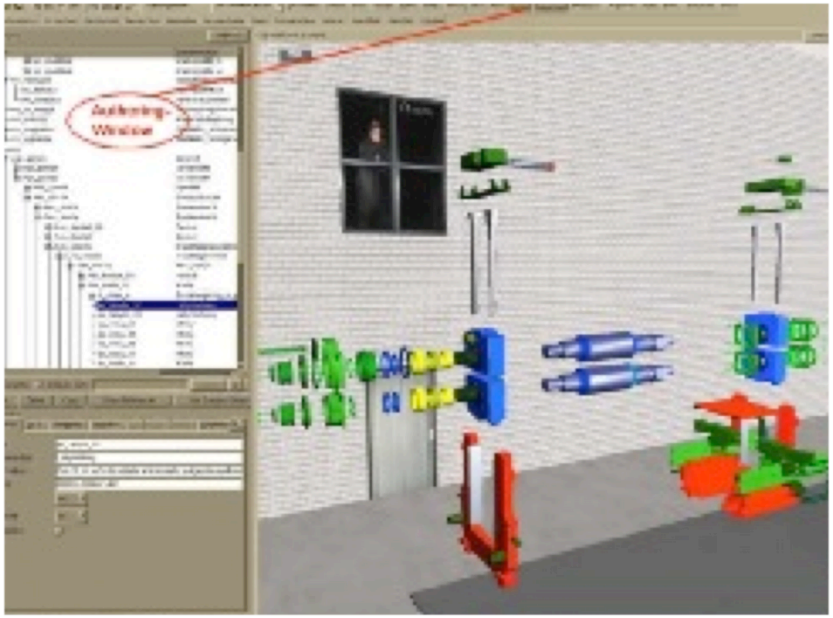
- **Procedure or Decision** training simulator [Ponder et al. 2003]:
  - Emphasizes on training procedure execution;
  - Disregards the evaluation of the user's dexterity;
  - Low user interaction requirements.
- Most common applications:
  - Maintenance and operation of complex machinery;
  - Life-threatening tasks;
  - Training takes place on virtual prototypes (physical device is unavailable).

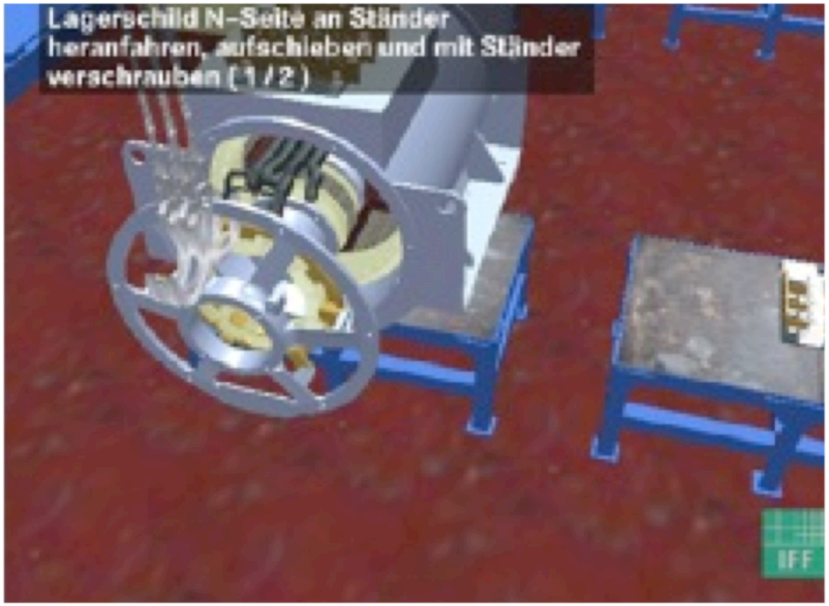
Introduction

- **Goals:**
  - Allow virtual training activities to be deployed and executed everywhere – desktop, mobile phones and tablets
  - Maintainability and creation of new training scenarios
    - Allow re-usability
- Our approach:
  - Uses X3D and X3DOM as the foundation for a standard web-based solution
  - Augment X3D by presenting additional nodes which facilitates the creation and maintenance of procedure training activities

Related work

- VDT Platform [Blümel et al. 2005]
  - Fraunhofer IFF
  - Generic frameworks to facilitate the creation of procedure training scenarios
  - GUI oriented designer
  - Developed in C++, not targeting web applications.
  - Our approach adopt Blümel's taxonomy





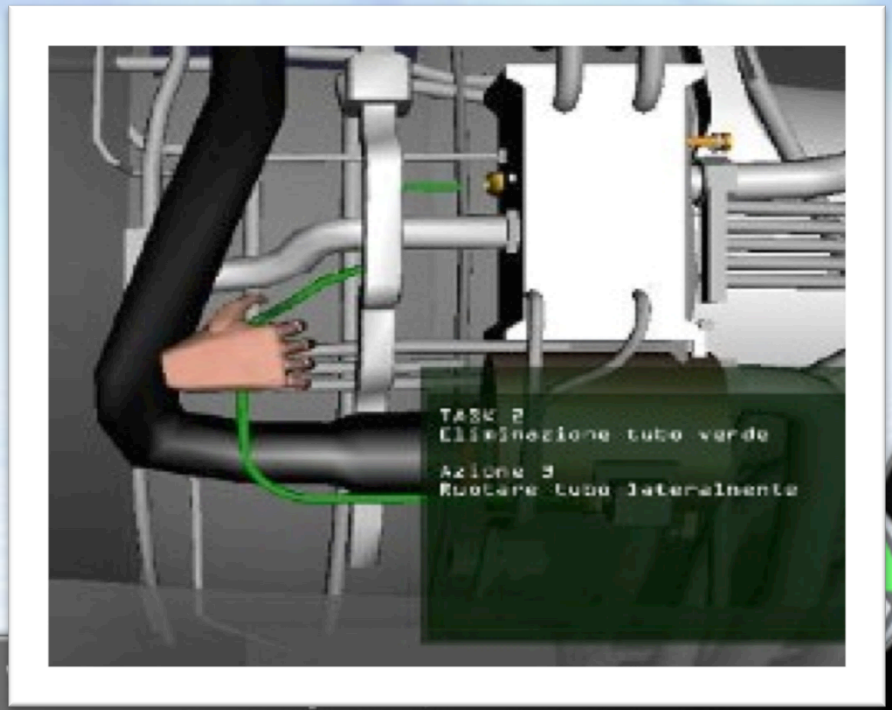
Related work

- Project GVT [Gerbaud et al. 2008]
  - INRIA
  - Another generic framework
  - Applications on military equipment
    - More than 50 procedures
  - Also developed in C++

- Similar commercial tools:
  - EON Reality and Virtools.

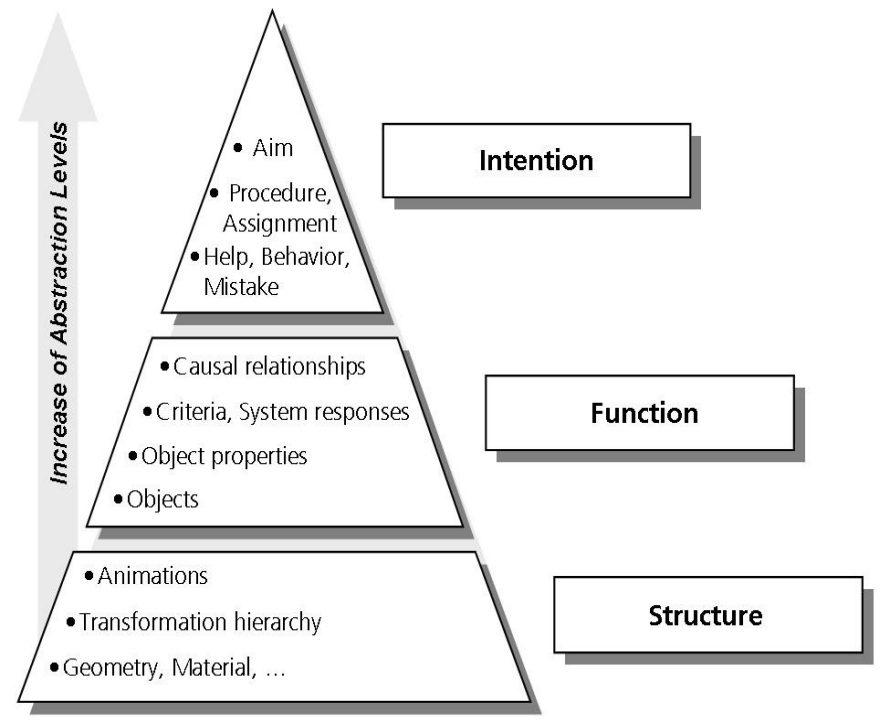
Related work

- Medical simulator in X3D by [Jung et al. 2008]
  - Targeted at training motor and dexterity skills;
  - There is no emphasis on procedure or decision training.
- iVT training tool by [Corvaglia 2004]
  - Allows the creation of procedure training scenarios by custom VRML nodes;
  - Very similar to our approach;
  - However, details are not clearly unveiled;
  - Does not support web-based applications.



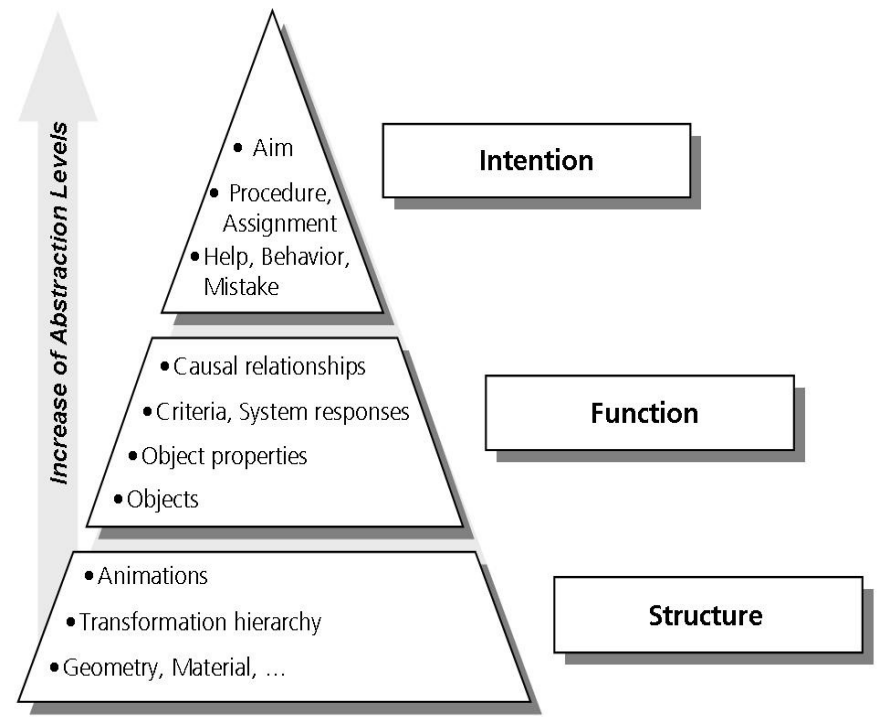
Procedure Training Simulators

- Our approach follows the taxonomy proposed by [Blümel et al. 2005]
  - The standard X3D nodes are used to define the Structure layer;
  - New nodes are implemented to meet the requirements of the topmost layers;



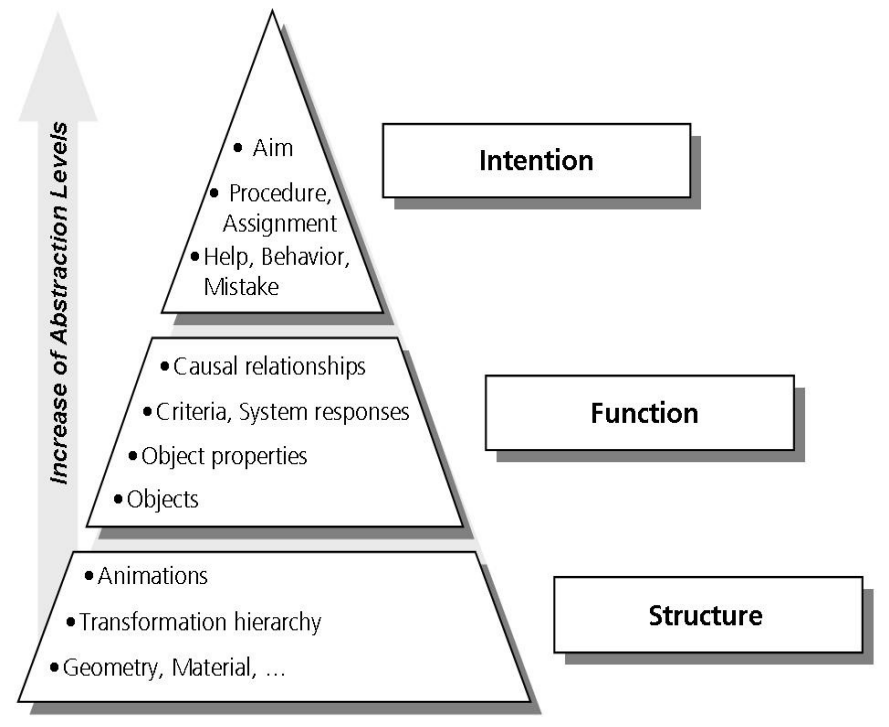
Procedure Training Simulators

- Middle layer - Defines the interactive objects in the scene and their behaviors
- Most common approaches used to define behavioral objects [Mollet et al. 2007]:
  - Sensor actuator-networks;
  - Rule-based systems;
  - Finite state machine;
- Our approach does allows different techniques to be used
  - Implementation was focused on FSMs



Procedure Training Simulators

- Topmost layer – Defines the procedure to be executed by the user
- Common alternatives to define dependency relationships:
  - Goals and Constraints [Wasfy et al. 2004];
  - Graphs [Ponder et al. 2003];
  - LORA Diagram [Gerbaud et al. 2008];
  - Petrinets [Lin et al. 2002];
- Our approach allows different techniques to be used
  - Implementation was focused on Petrinets



Proposed Nodes

Nodes for Procedure Simulators

- Abstract **TrainingObjectNode**
  - Represents the second layer on Blümel taxonomy;

```
TrainingObjectNode : X3DGroupingNode {
        MFNode [out] actions : TrainingObjectActionNode
}
```

  - Group scene elements in it's *children* attribute
  - *actions*: Attribute with the object's available actions
  - Specializations should determine behavior

Nodes for Procedure Simulators

- Abstract **TrainingObjectActionNode**
  - The only way to interact with objects in the scene
  - User must choose which action to execute from the list of available actions

```
TrainingObjectActionNode : X3DNode {
        SFBool  [in]      execute
        SFBool  [out]     executed
        SFTime  [out]     executedTime
}
```

  - Output attributes allow events to be routed
    - Trigger animations to reproduce the action's effects

Nodes for Procedure Simulators

- **StateMachineObject**
  - Specialization of **TrainingObjectNode**
  - Object behavior is defined by a Finite State Machine

```
StateMachineObject : TrainingObjectNode {
        MFString []      url
        MFNode  [in]     transitionActions :
StateMachineTransitionAction
}
```

  - *url*: Attribute pointing to file with FSM description
  - *transitionActions*: Input attribute with actions which are represented by transitions of the state machine.

  - Behavior: Select the available actions from *transitionActions* and place them on the output attribute *actions*.

Nodes for Procedure Simulators

- Abstract **TrainingProcedureNode**
  - Represents the top most layer on Blümel taxonomy;

```
TrainingProcedureNode : X3DBindableNode {
        MFNode [out]  forwardActions     :  TrainingObjectActionNode
        MFNode [out]  backwardActions    :  TrainingObjectActionNode
        SFBool     [in]    executeForward
        SFBool     [in]    executeBackward
        SFNode   [out]  executedAction     :  TrainingObjectActionNode
        SFBool     [out]  isFinished
}
```

  - Provides the next **correct** actions for the current procedure
    - It **knows** all the actions from all the existing objects in the scene
  - Specializations should determine the procedure definition

Nodes for Procedure Simulators

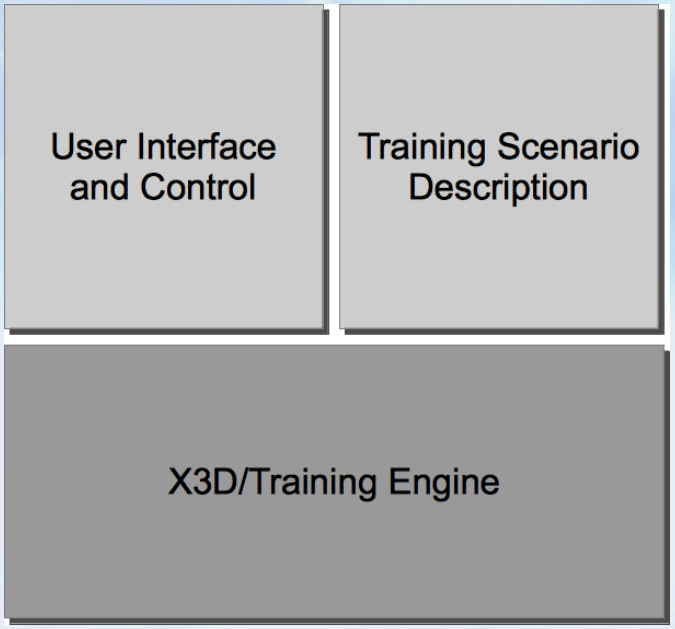
- **PetrinetProcedure**
  - Specialization of **TrainingProcedureNode**
  - A *transition* in a Petrinet represents an object action

```
PetrinetProcedure : TrainingProcedureNode {
        MFString []     url
}
```

  - *url*: Attribute pointing to file with Petrinet description
  - Behavior: Check the available transitions in the Petrinet and places the associated actions in its output attributes.

Implementation

- Nodes implemented directly into X3DOM
  - No PROTOs available yet
- Motivations:
  - Support on most modern browsers;
    - Requires WebGL
  - Create applications easily:
    - X3DOM + HTML5 + Javascript



Workflow

- Overview of a Training Scenario Workflow:
  - Create an X3D scene: Geometric models and animations
  - Create interactive objects by grouping their geometric elements into **StateMachineObject** node's
    - Create a FSM for each object
    - Insert their actions (transitions of their FSM)
    - ROUTE those actions to appropriate animations
  - Create a training procedure (e.g. Petrinet) and place a **PetrinetProcedure** node in the scene.
    - Create Petrinet with PNML file format
    - Associate the Petrinet transitions with the objects actions
  - Insert the required meta-data to hold additional information
    - Part names, functional description, icons, attached files and others – Might be used for instruction or create the application GUIs.

Validation

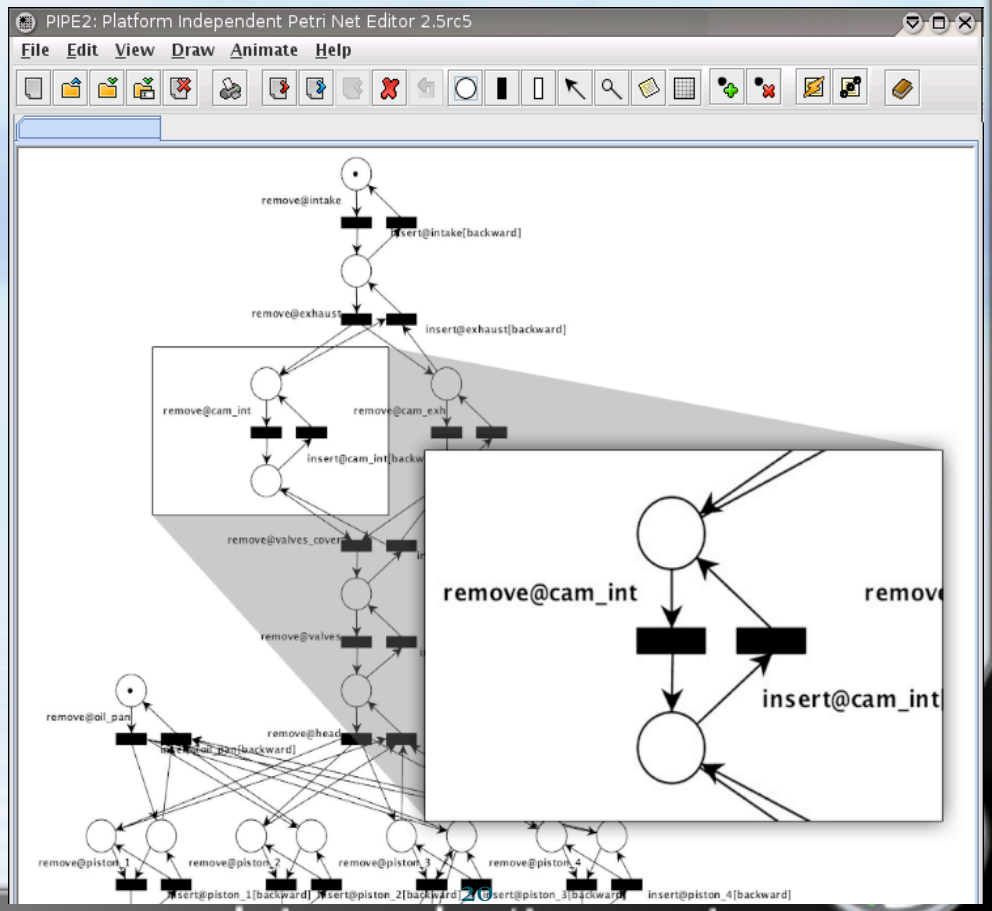
Finite State Machine

- Currently supports a subset of SCXML

```
<?xml version="1.0" encoding="UTF-8"?>
<scxml xmlns="http://www.w3.org/2005/07/scxml" version="1.0" initial="inserted">
  <state id="inserted">
    <transition event="remove" target="removed"/>
  </state>
  <state id="removed">
    <transition event="insert" target="inserted"/>
  </state>
</scxml>
```

Petrinet

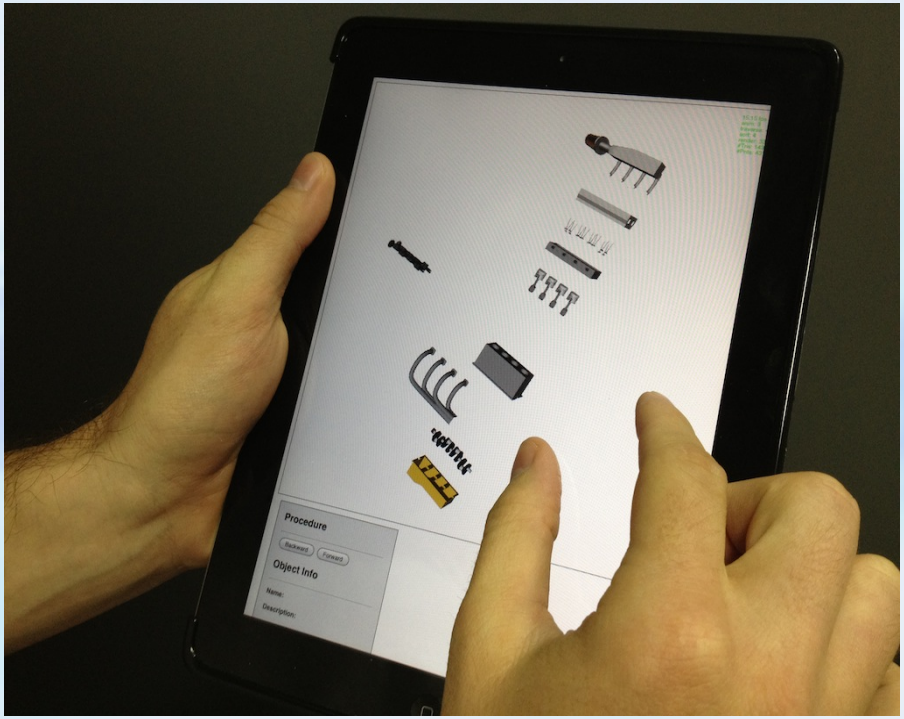
- Supports PNML file format



Results

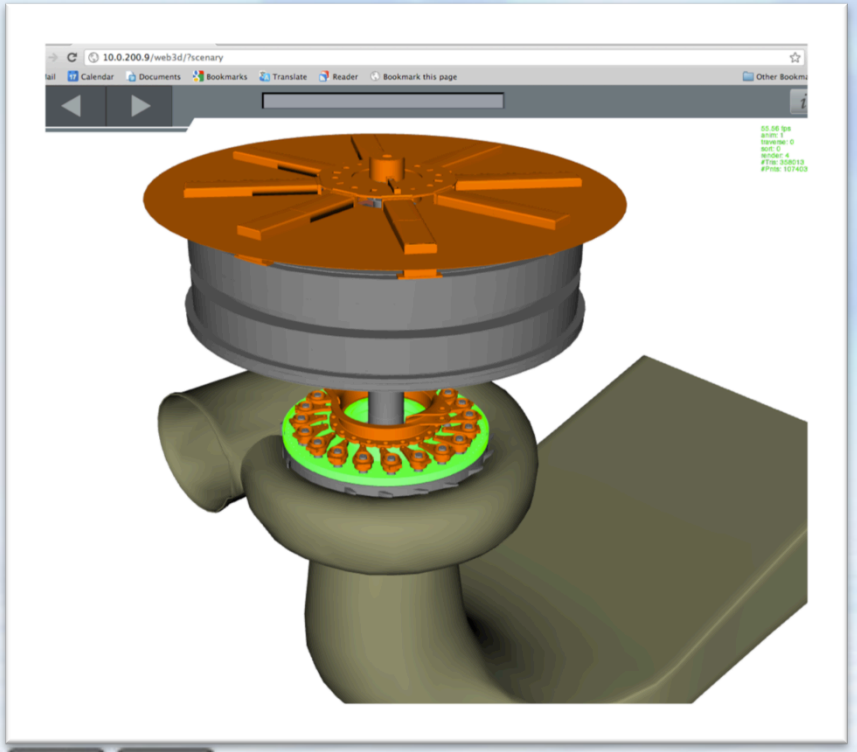
Results

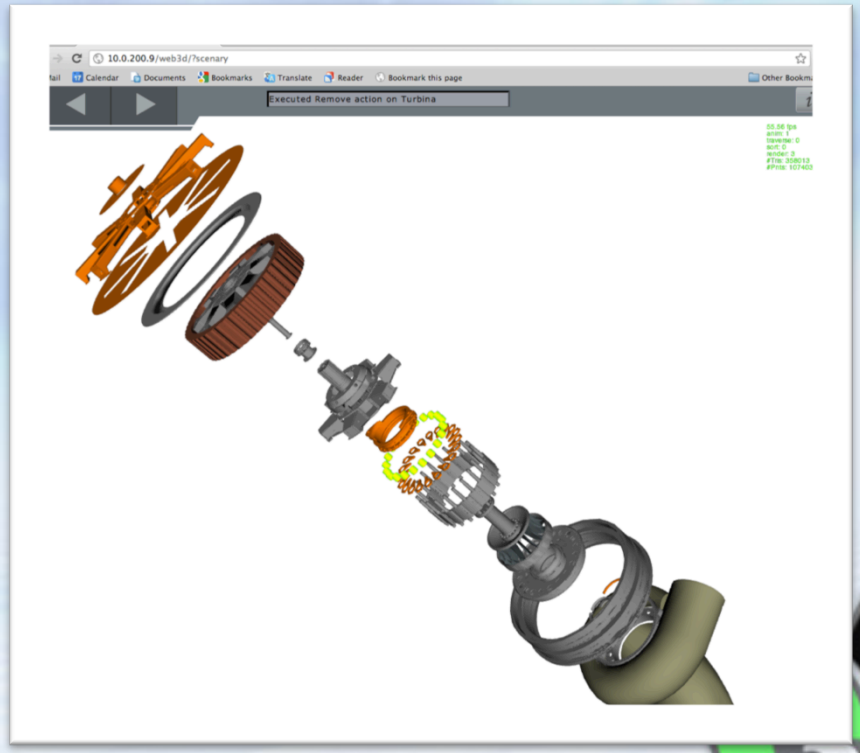
- Training Application
  - User selects an object
  - Available actions are displayed
  - User selects the desired action
  - The application checks if the action is correct:
    - Ok, the action is executed
    - Otherwise, an error message is displayed
  - The user can also request the application to automatically advance or retrocede the procedure



Use Cases

- Real assembly-disassembly procedure
  - Hydroeletric Generating Unit;
  - Large geometric model;
  - Simple training procecure - Sequential;

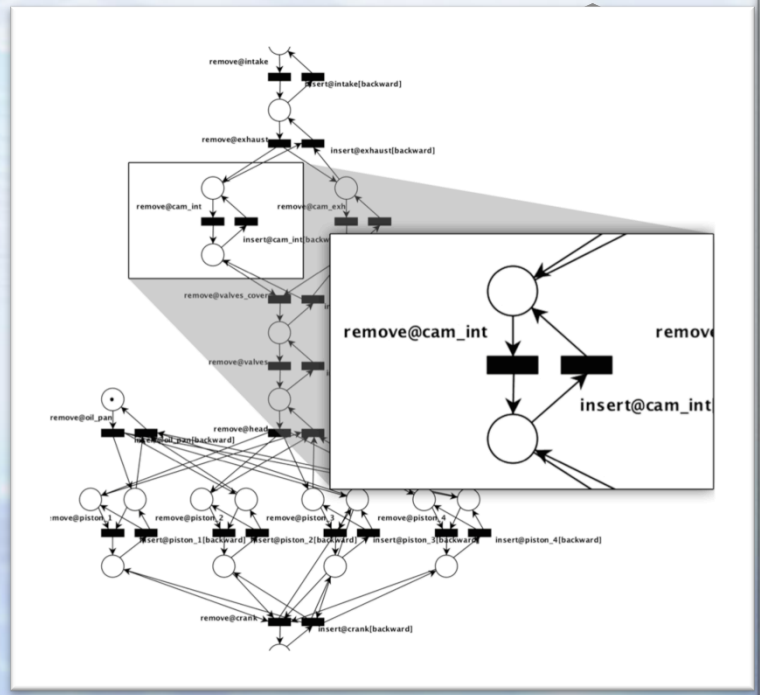


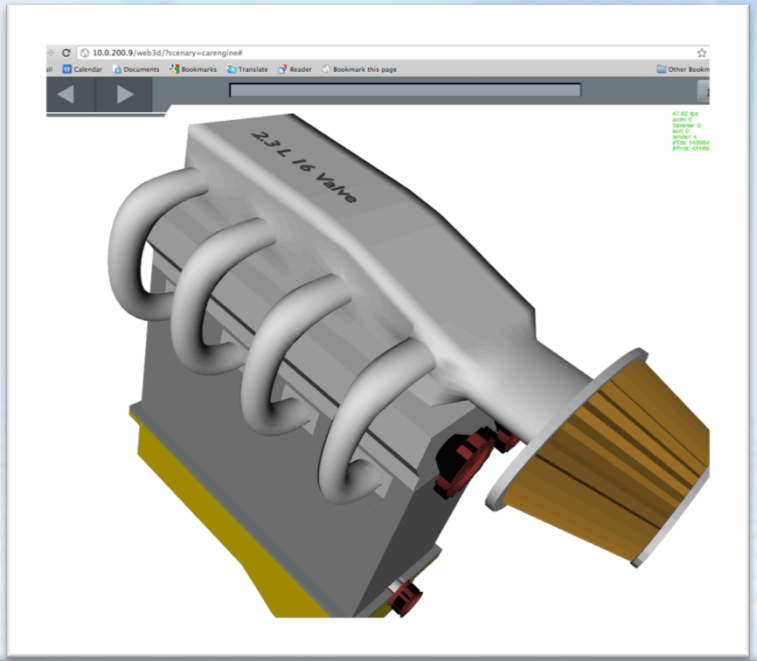


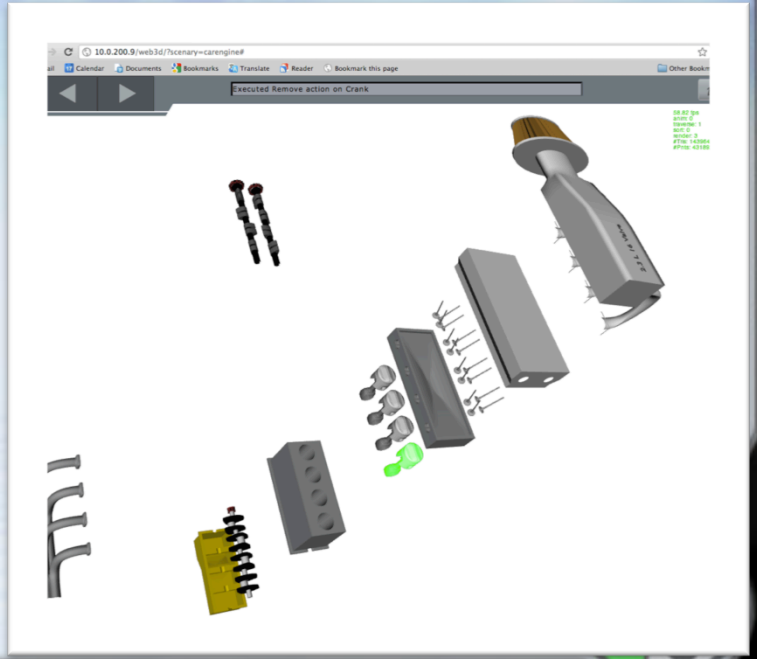
23

Use Cases

- Fictitious procedure
  - Car engine disassembly
  - Simple geometric model
    - Mobile devices
  - More complex procedure







Conclusions

- Architecture for creating procedure training simulators in X3D
  - Declarative syntax
  - Easier path for end users
  - Step closer to easily create/edit training applications
    - WebGL + HTML5 + Javascript
  - From desktop to mobile
  - Requires only a web browser

Limitations

- Manually editing the X3D scene files can be labor intensive
  - Embed much of the required workflow into plugins to 3D modeling tools;

- We believe our approach can be used as a guideline for authoring tools

References

- BEHR, J., ESCHLER, P., JUNG, Y., AND ZÖLLNER, M. 2009. X3dom: a dom-based html5/x3d integration model. In *Proceed- ings of the 14th International Conference on 3D Web Technol- ogy,* ACM, New York, NY, USA, Web3D ' 09, 127–135.
- BLÜMEL, E., MÜLLER, G., SALEM, W., AND SCHENK, M. 2005. Technology enhanced training at workplace: A virtual reality based training system for the technical domain. In *Proceed- ings of the 1st International Conference on E-Business and E- Learning, EBEL,* 57–62.
- CORVAGLIA, D. 2004. Virtual training for manufacturing and maintenance based on web3d technologies. In *Proc. of the 1st In- ternational Workshop on Web3D Technologies in Learning, Ed- ucation and Training,* LET-WEB3D 2004, 28–33.
- CREMER, J., KEARNEY, J., AND PAPELIS, Y. 1995. HCSM: A framework for behavior and scenario control in virtual environ- ments. *ACM Transactions on Modeling and Computer Simula- tion 5,* 242–267.
- DEV, P., YOUNGBLOOD, P., HEINRICHS, W. L. H., AND KUSUMOTO, L. 2007. Virtual worlds and team training. *Anes- thesiology Clinics 25,* 321–336.
- EON-REALITY. visited on June 2012. http://eonreality.com.
- GALLAGHER, A. G., RITTER, E. M., CHAMPION, H., HIGGINS, G., FRIED, M. P., MOSES, G., SMITH, C. D., AND SATAVA, R. M. 2005. Virtual reality simulation for the operating room - proficiency-based training as a paradigm shift in surgical skills training. *Annals of Surgery 241(2)* (February), 364–372.
- GERBAUD, S., MOLLET, N., GANIER, F., ARNALDI, B., AND TISSEAU, J. 2008. GVT: a platform to create virtual environ- ments for procedural training. In *IEEE Virtual Reality Confer- ence,* 225–232.
- JUNG, Y., RECKER, R., OLBRICH, M., AND BOCKHOLT, U. 2008. Using x3d for medical training simulations. In *Proceed- ings of the 13th international symposium on 3D web technology,* ACM, New York, NY, USA, Web3D ' 08, 43–51.
- LIN, F., YE, L., DUFFY, V. G., AND SU, C.-J. 2002. Developing virtual environments for industrial training. *Inf. Sci. Inf. Comput. Sci. 140,* 1, 153–170.
- WEB 3D CONSORTIUM, 2012. X3d script node specification iso- iec-19775-x3dabstractspecification, April.
- MOLLET, N., GERBAUD, S., AND ARNALDI, B. 2007. STORM: a generic interaction and behavioral model for 3d objects and humanoids in a virtual environment. In *IPT-EGVE the 13th Eu- rographics Symposium on Virtual Environments,* 95–100.
- PONDER, M., HERBELIN, B., MOLET, T., SCHERTENLIEB, S., ULICNY, B., PAPAGIANNAKIS, G., MAGNENAT-THALMANN, N., AND THALMANN, D. 2003. Immersive vr decision train- ing: telling interactive stories featuring advanced virtual human simulation technologies. In *EGVE '03: Proceedings of the work- shop on Virtual environments 2003,* ACM, New York, NY, USA, 97–106.
- RICKEL, J., AND JOHNSON, W. L. 1998. Animated agents for procedural training in virtual reality: Perception, cognition, and motor control. *Applied Artificial Intelligence 13,* 343–382.
- SOPIN, I., AND HAMZA-LUP, F. G. 2010. Extending the web3d: design of conventional gui libraries in x3d. In *Proceedings of the 15th International Conference on Web 3D Technology,* ACM, New York, NY, USA, Web3D ' 10, 137–146.
- VAN DAM, A. 1997. Post-wimp user interfaces. *Commun. ACM 40,* 2 (Feb.), 63–67.
- VAN DE PANNE, M., AND FIUME, E. 1993. Sensor-actuator net- works. In *SIGGRAPH '93: Proceedings of the 20th annual con- ference on Computer graphics and interactive techniques,* ACM, New York, NY, USA, 335–342.
- VIRTOOLS. visited on June 2012. http://www.3ds.com/products/3dvia/3dvia-virtools.
- WASFY, A., WASFY, T., AND NOOR, A. 2004. Intelligent virtual environment for process training. *Adv. Eng. Softw. 35,* 6, 337– 355.
- WEBER, M., AND KINDLER, E. 2003. The petri net markup lan- guage. In *Petri Net Technology for Communication-Based Sys- tems,* H. Ehrig, W. Reisig, G. Rozenberg, and H. Weber, Eds., vol. 2472 of *Lecture Notes in Computer Science.* Springer Berlin / Heidelberg, 124–144.

Thank you!

- Questions ?

- Contact:

mcabral@lsi.usp.br

www.lsi.usp.br/~mcabral