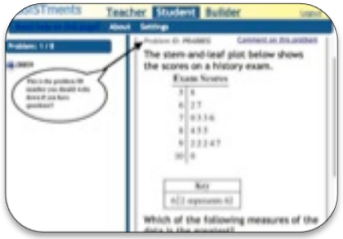
Towards ITS Authoring Tools for Domain Experts

Robert Taylor
Andy Smith
Bradford Mott
Samuel Leeman-Munk
James Lester

Department of Computer Science
North Carolina State University

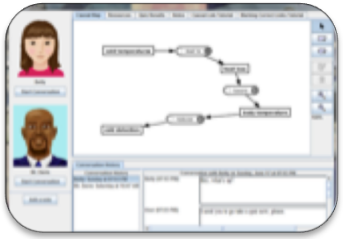
Authors

Authoring Tools



Assistments

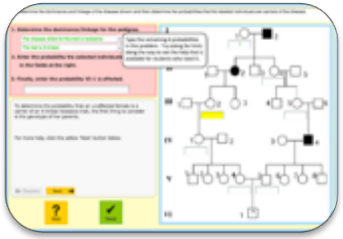
AutoTutor



Betty's Brain

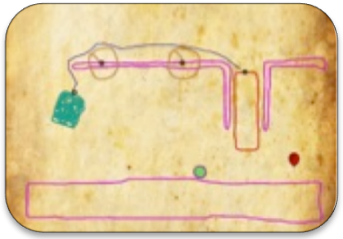
Bio-World

Crystal Island



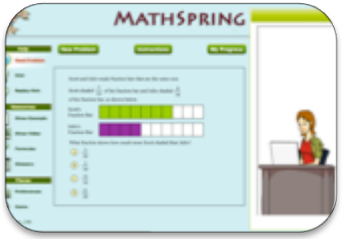
Genetics Tutor

GIFT - vMedic



Physics Playground

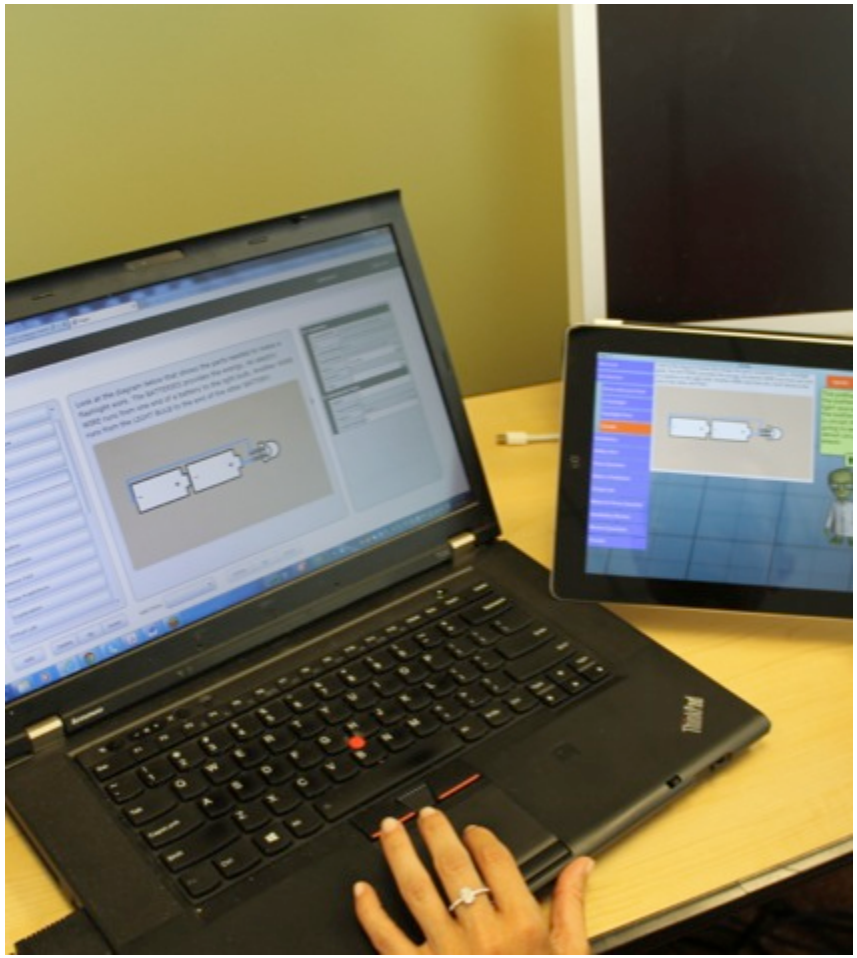
Virtual Cultural Awareness Trainer



Wayang Math Tutor

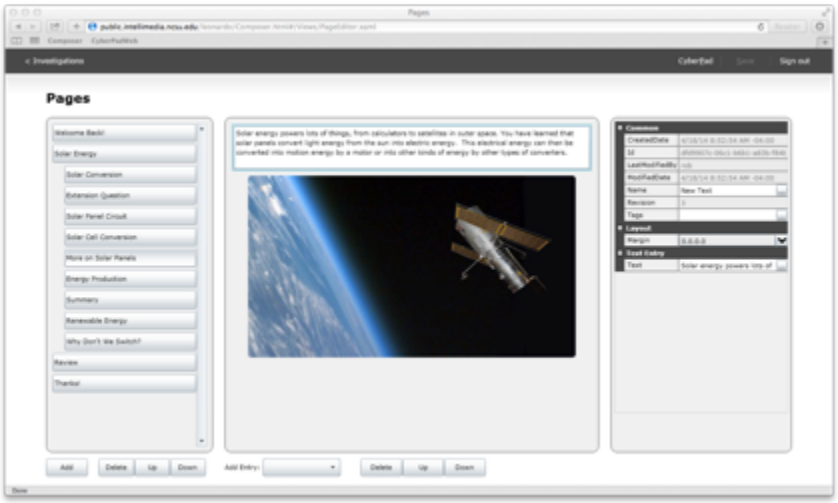
Challenges of ITS Authoring

- Authors who are not ITS experts
- Enabling collaboration
- Reducing the ITS complexity exposed to the author



Design Principles for ITS Authoring Tools

- Design principles derived from our experience
- Adopting existing UI and workflows
- Leveraging software engineering techniques



Outline

- Related Work
- Case Study: Authoring the LEONARDO Digital Science Notebook
- Design Principles for ITS Authoring Tools
- Conclusion and Future Work

Outline

- Related Work
- Case Study: Authoring the LEONARDO Digital Science Notebook
- Design Principles for ITS Authoring Tools
- Conclusion and Future Work

Related Work

Eon [Murray 1997] is an authoring tool prototype for specifying:

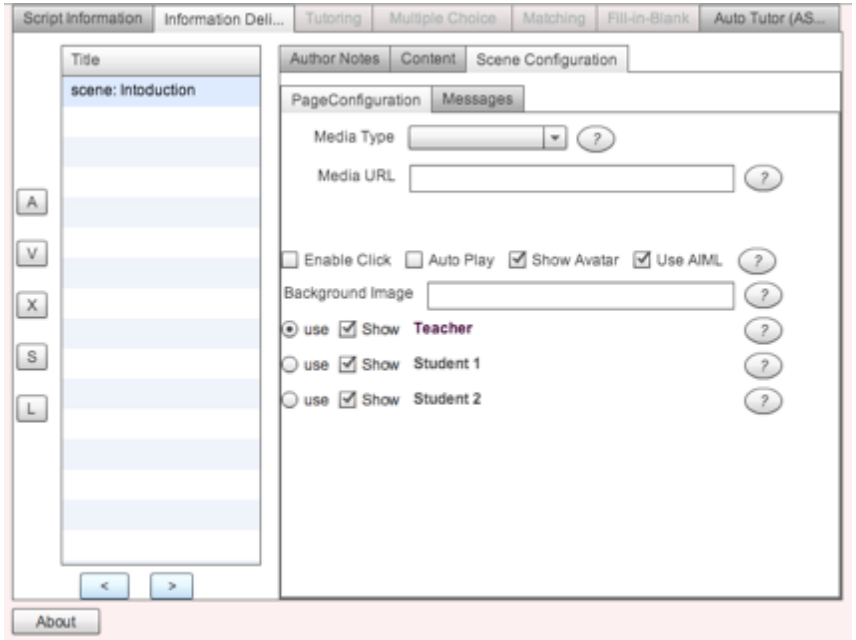
- Learning environment
- Domain knowledge
- Teaching Strategies
- Student Model

Eon

Related Work

ASAT [Graesser 2005] is used to build AutoTutor-based tutors:

- Conversation scripts
- Content
- Production rules

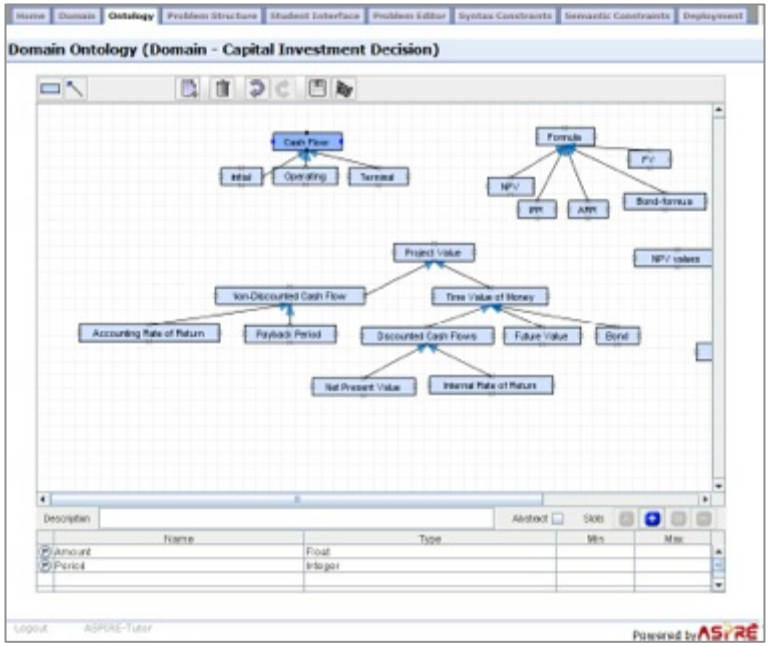


AutoTutor Script Authoring Tool (ASAT)

Related Work

ASPIRE [Mitrovic 2009] is a web-based tool for creating constraint-based tutors:

- Model problems & solutions
- Design student interface
- Authoring-server
- Deployment-server

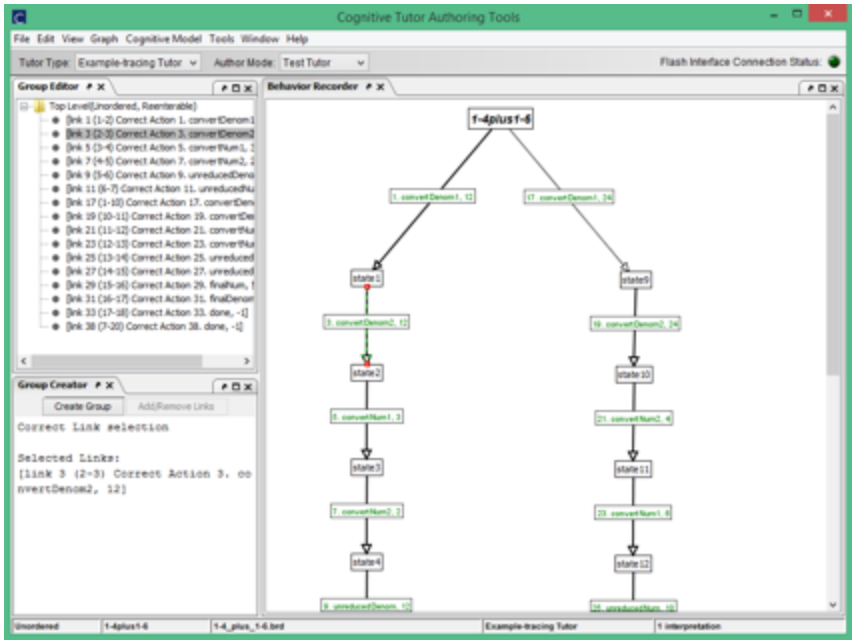


ASPIRE

Related Work

CTAT [Aleven 2009] facilitates the development of cognitive tutors:

- Example-tracing tutors
- "Mass Production" to speed up content creation
- Used by non-programmers



Cognitive Tutor Authoring Tools (CTAT)

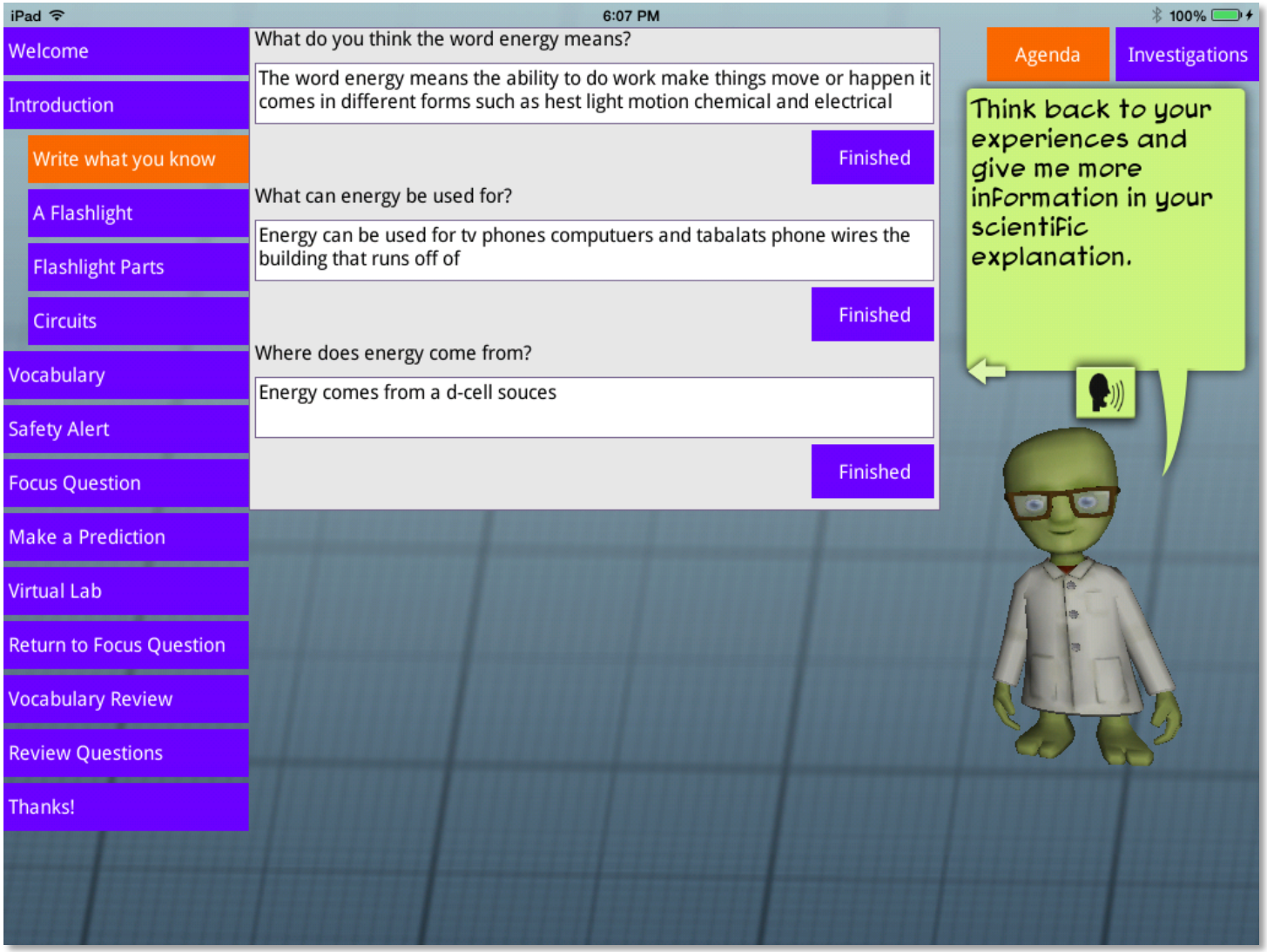
Outline

- Related Work
- Case Study: Authoring the the LEONARDO Digital Science Notebook
- Design Principles for ITS Authoring Tools
- Conclusion and Future Work

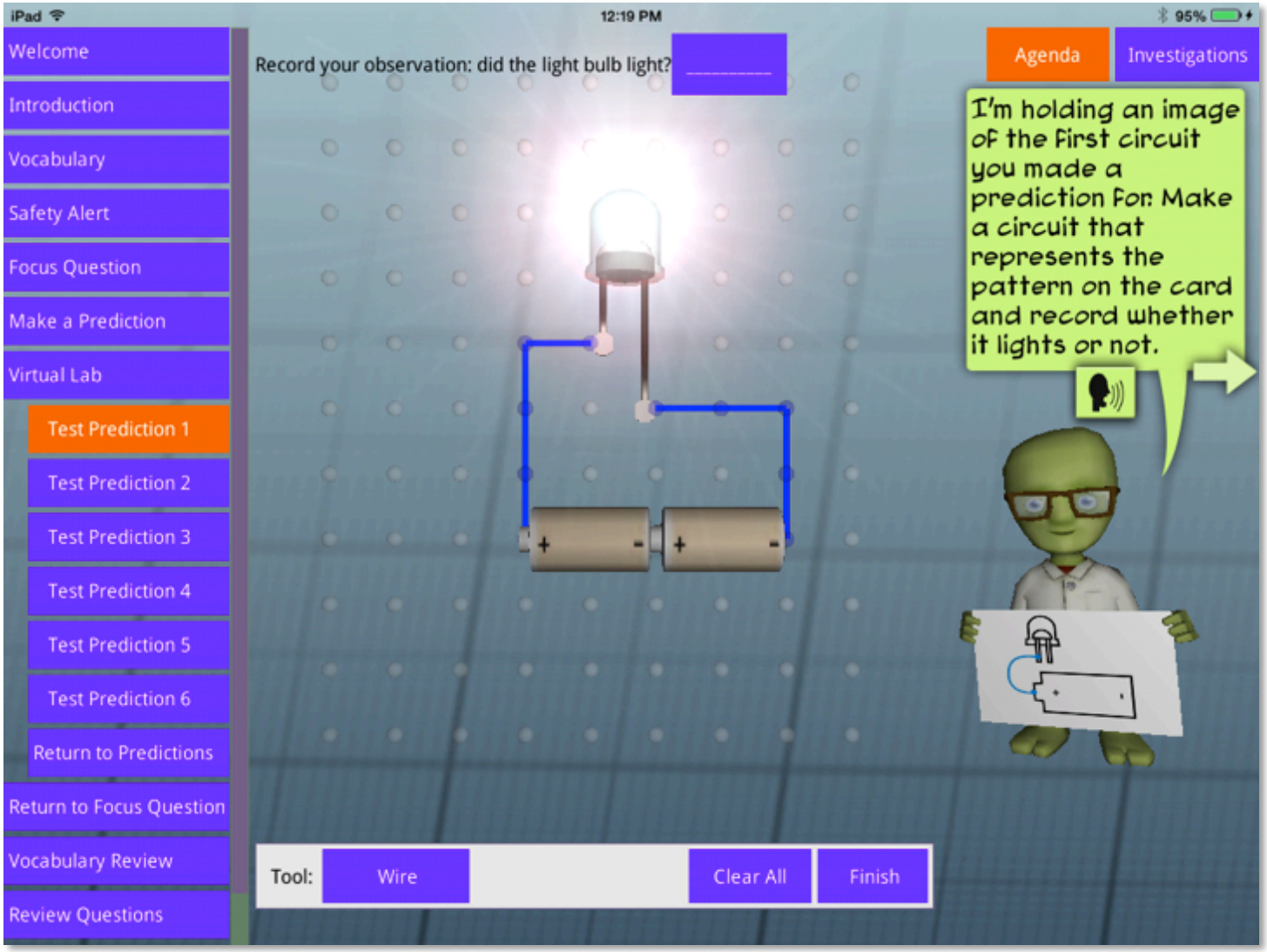
LEONARDO CyberPad Tablet Deployment

LEONARDO CyberPad Laptop Deployment

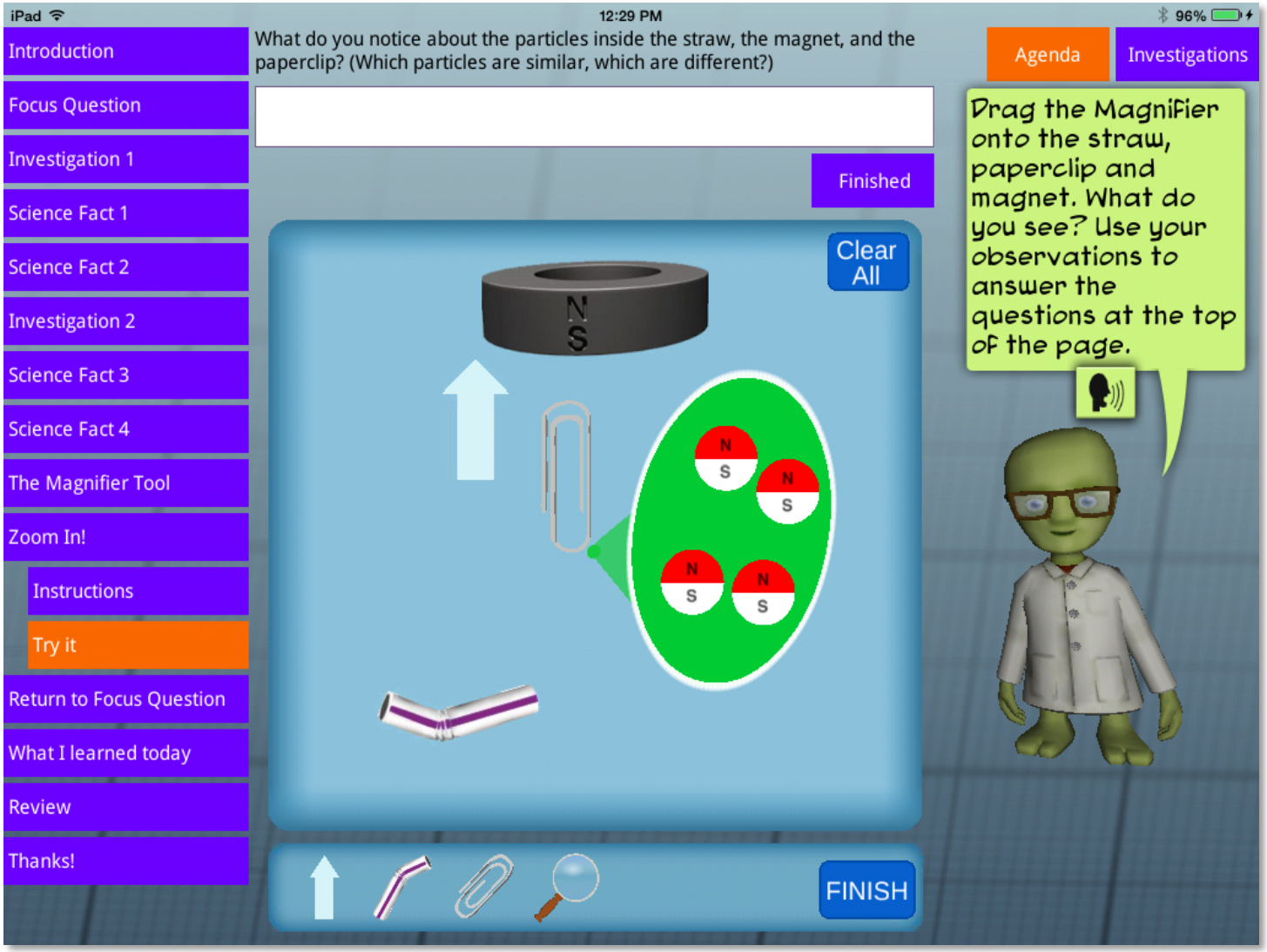
LEONARDO CyberPad



LEONARDO CyberPad



LEONARDO CyberPad



LEONARDO
Multi-State Deployment

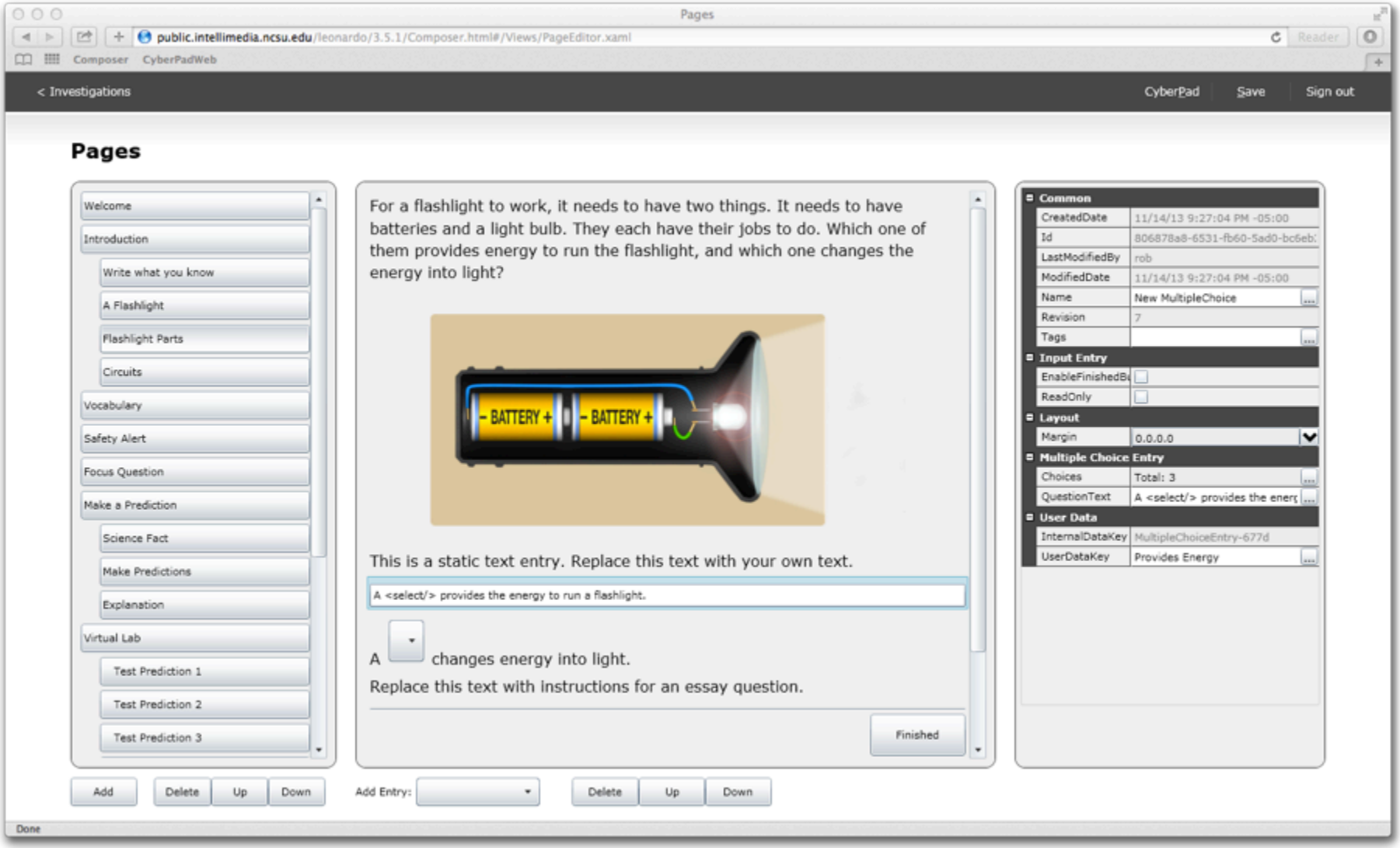
- States: 8
- Classrooms: 38
- Students: 912

LEONARDO Curricular Content

- Initial curricular content:
  - Energy & Circuits
  - Magnetism
- Based on Full Option Science System (FOSS)
- Includes physical and virtual experiments

LEONARDO Composer

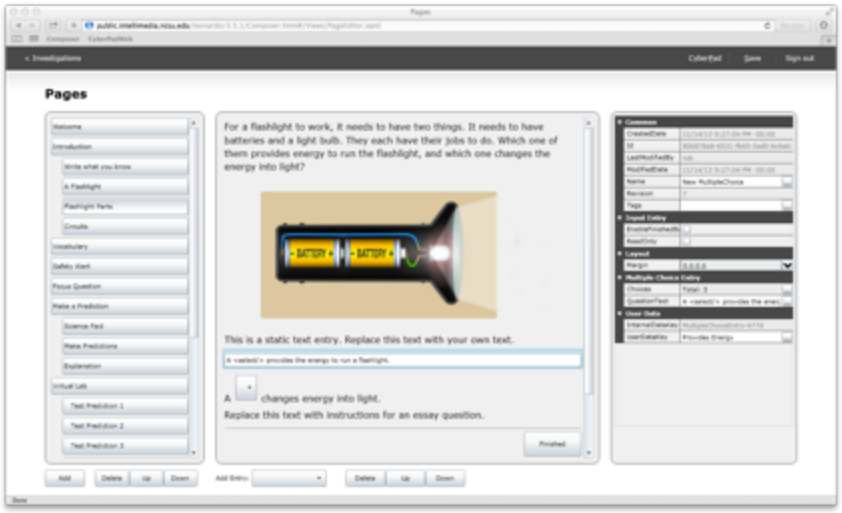
the intelliMEDIA group



LEONARDO Composer

Authors use Composer to create:

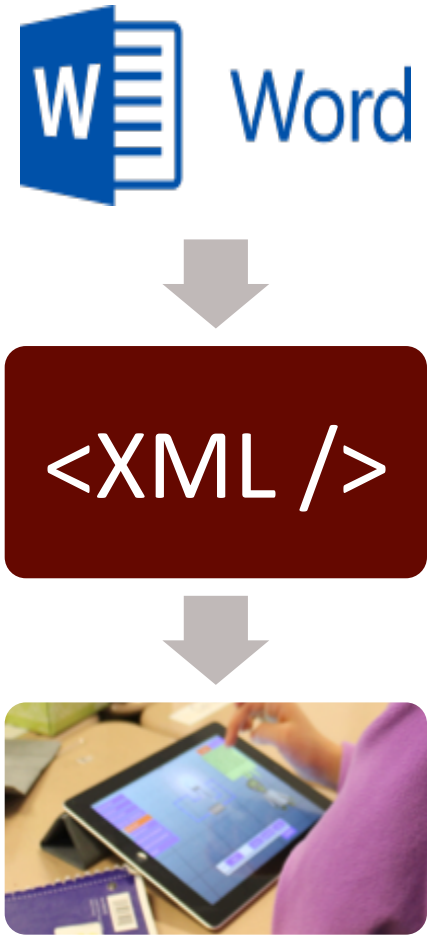
- Curricular content
- Agent dialogue
- Agent behavior



Prior to Composer

Original authoring workflow:

- Word used to author curriculum and agent dialogue
- Doc copied into XML
- XML embedded in the iPad app
- Agent behavior authored in code by software engineers



Prior to Composer

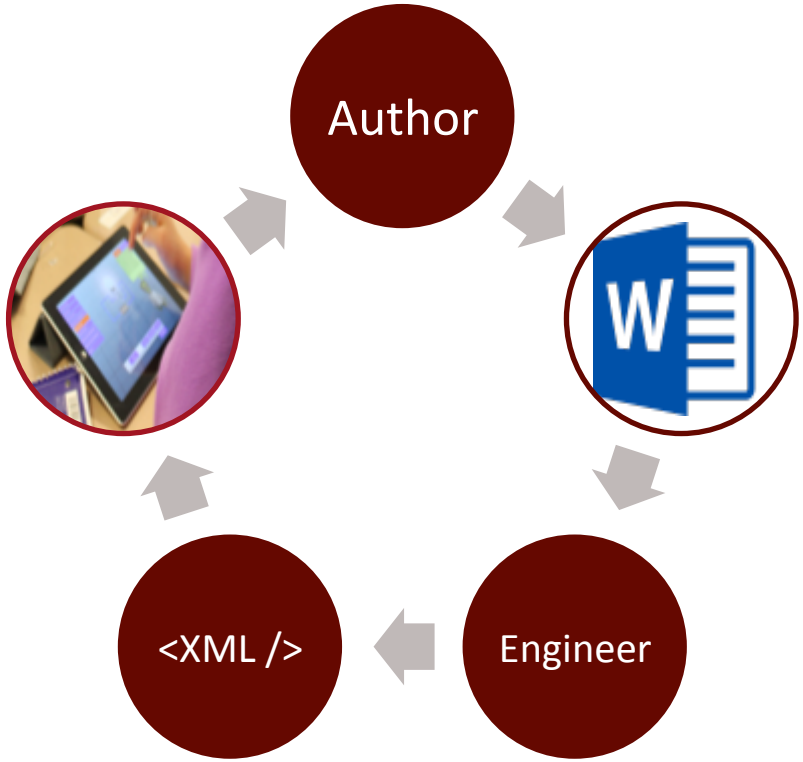
The following people were made sad by the original authoring workflow:

- John Bedward, *STEM Education*
- Courtney Behrle, *STEM Education*
- Michael Carter, *English*
- Kirby Culbertson, *Art and Design*
- Samuel Leeman-Munk, *Computer Science*
- Bradford Mott, *Computer Science*
- Lindsay Patterson, *STEM Education*
- Marc Russo, *Art and Design*
- Angela Shelton, *STEM Education*
- Ruth Sirkin, *STEM Education*
- Andy Smith, *Computer Science*
- Robert Taylor, *Computer Science*
- Eric Wiebe, *STEM Education*

Prior to Composer

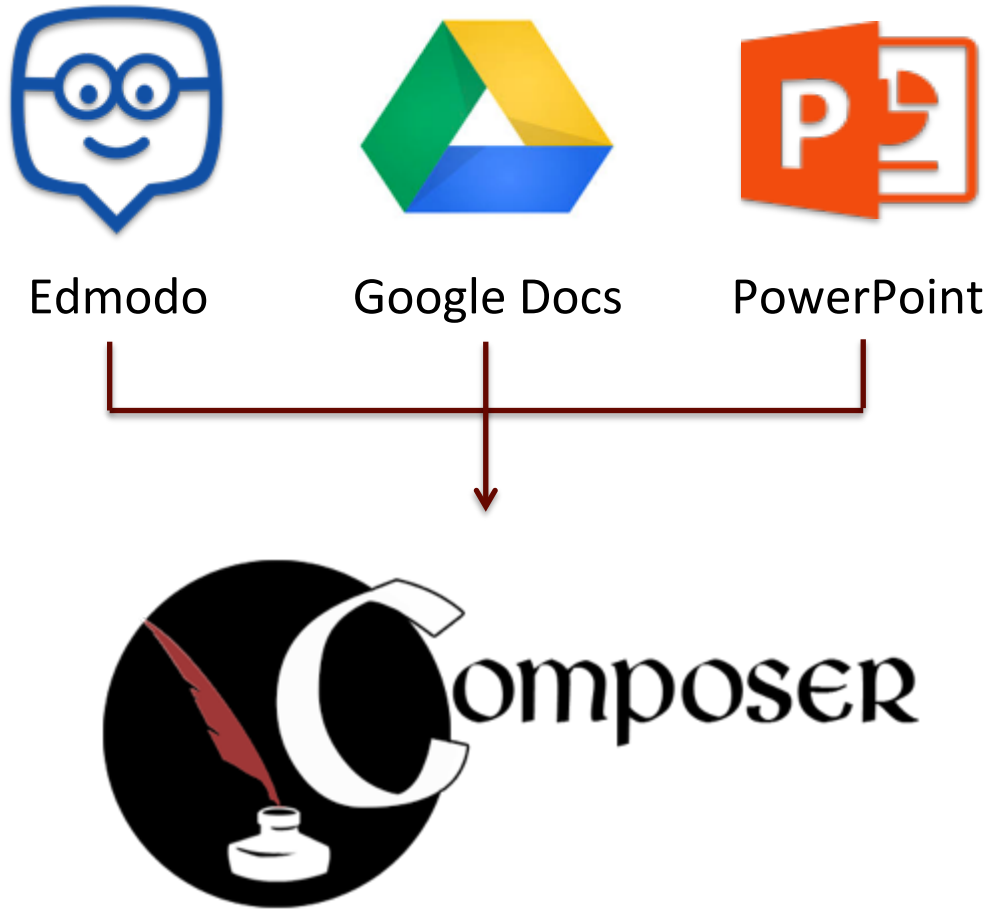
Drawbacks:

- 15+ minute iteration
- Lack of WYSIWYG
- “Collaboration” by e-mailing Word docs
- Prone to programmer error
- Rules authored in source code

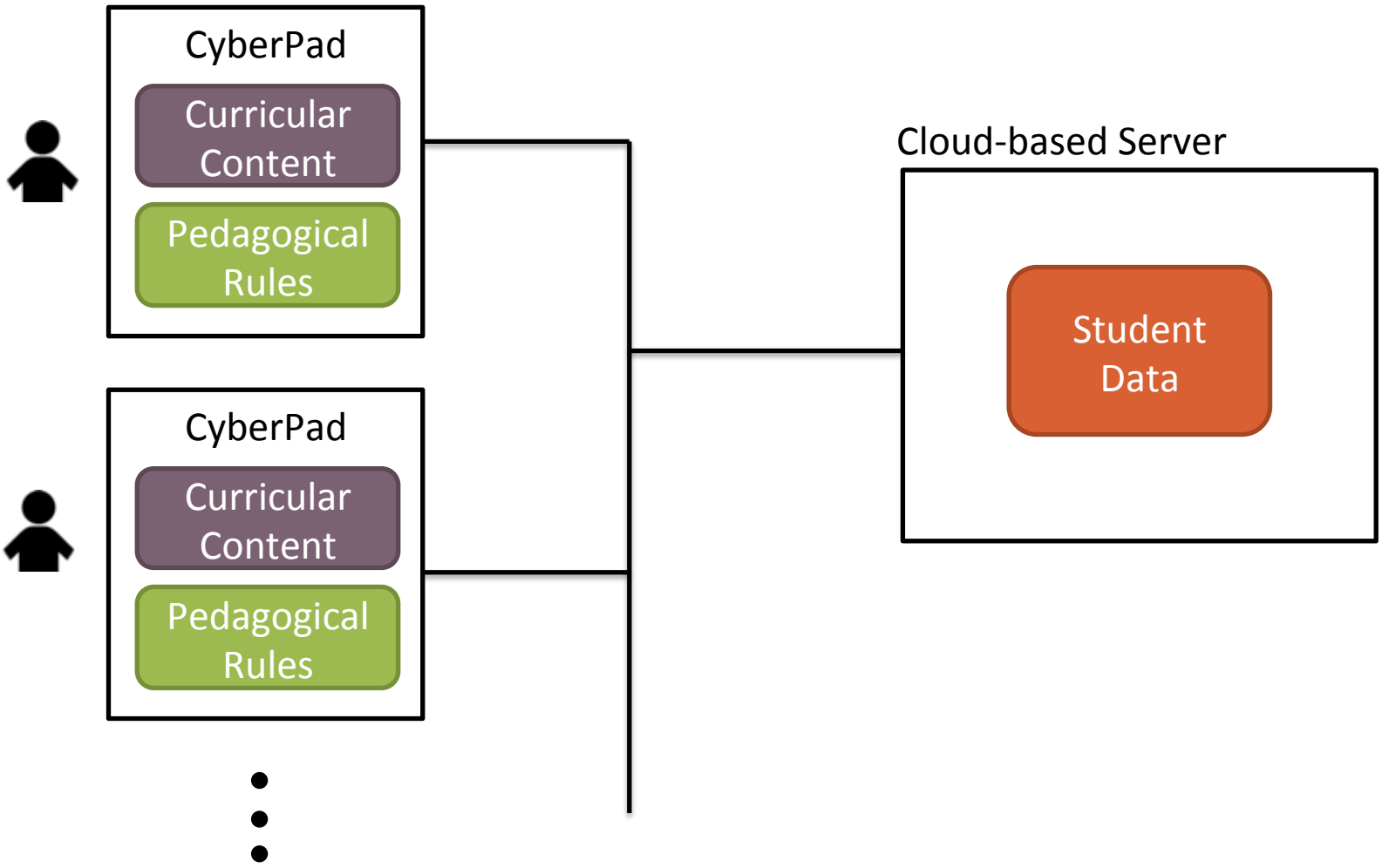


Lesson Learned: Create an Authoring Tool

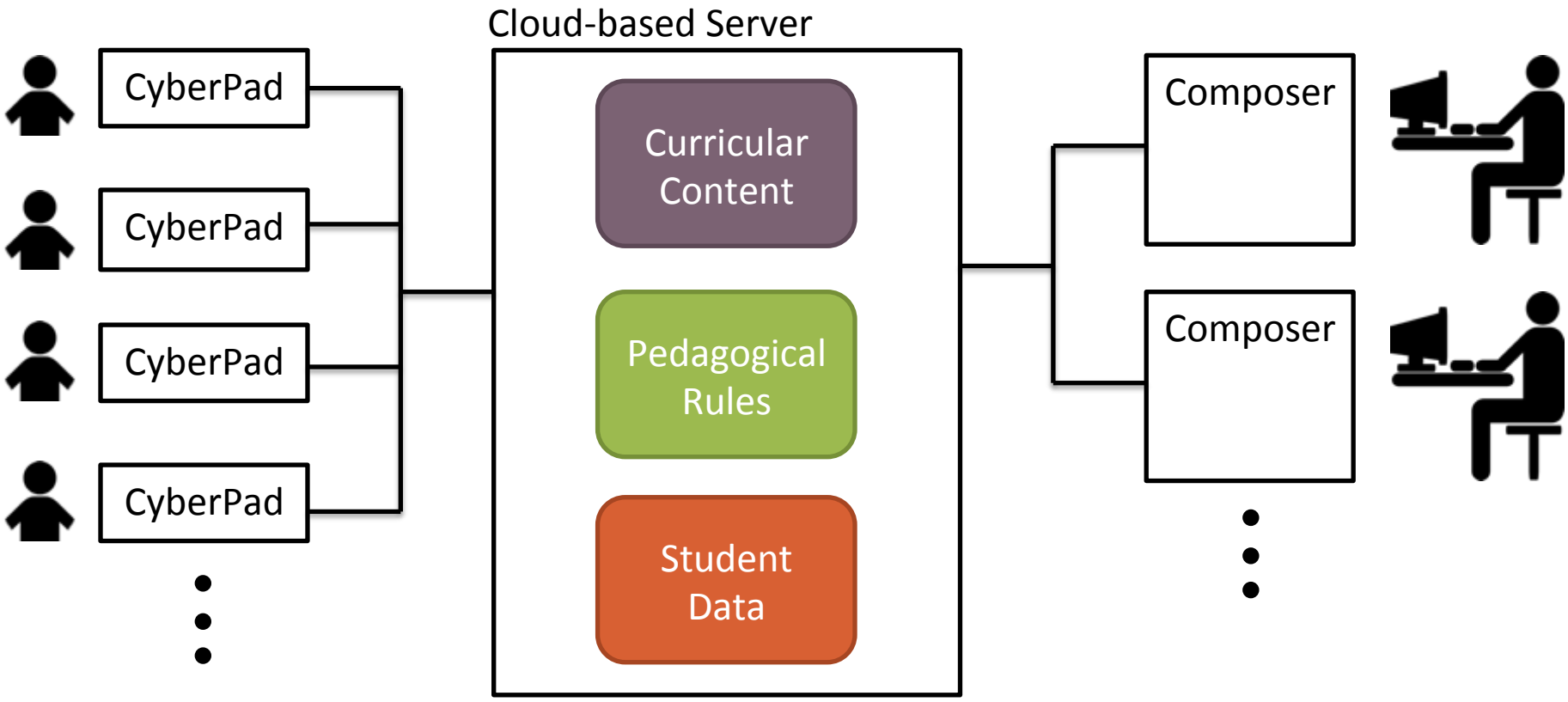
- Identify authors:
  - STEM experts
  - K-12 teachers (future)
- Identify familiar tools
- Design workflow based on familiar UIs & features
- Iteratively develop Composer based on author feedback



LEONARDO Architecture Prior to Composer

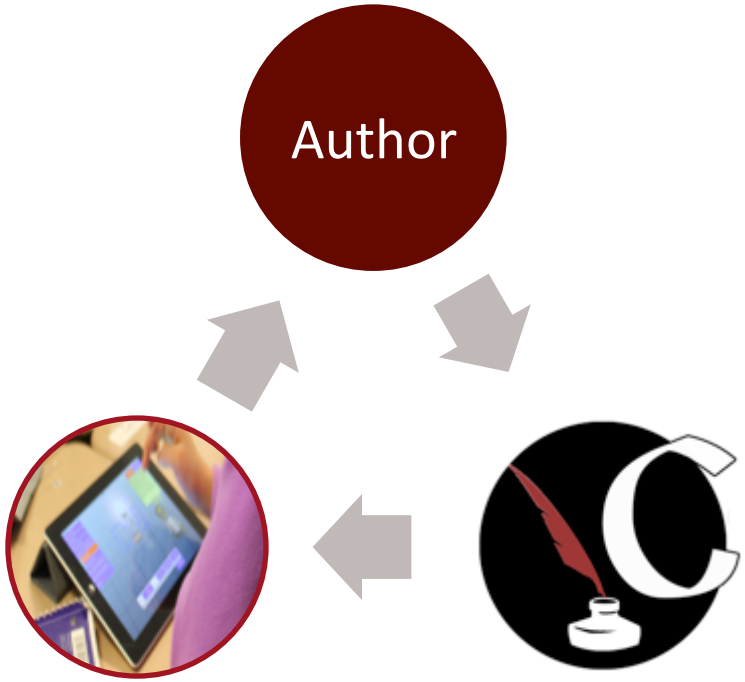


LEONARDO Architecture with Composer



Composer Features

- Simplified workflow
- Familiar UI
- Rapid iteration
- Curricular content stored in cloud
- Web-based authoring tool

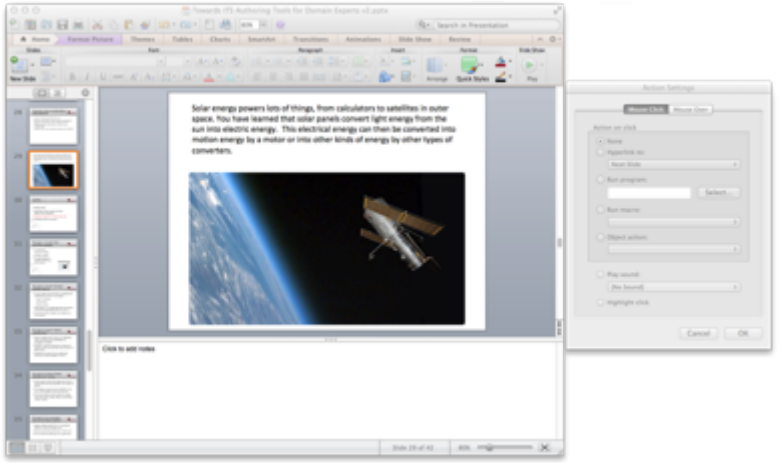


Outline

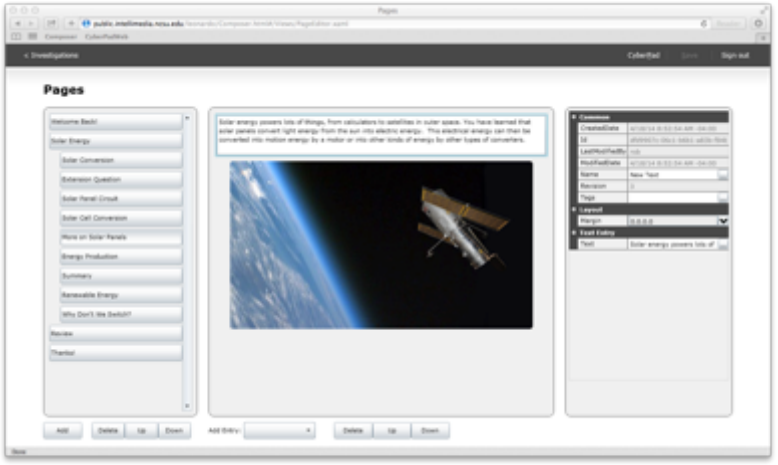
- Related Work
- Case Study: Authoring the LEONARDO Digital Science Notebook
- Design Principles for ITS Authoring Tools
- Conclusion and Future Work

Principle 1: Familiar User Interface Paradigm

- UI is the most important feature
- Familiar to author
- Leverage decades of usability and efficiency improvements



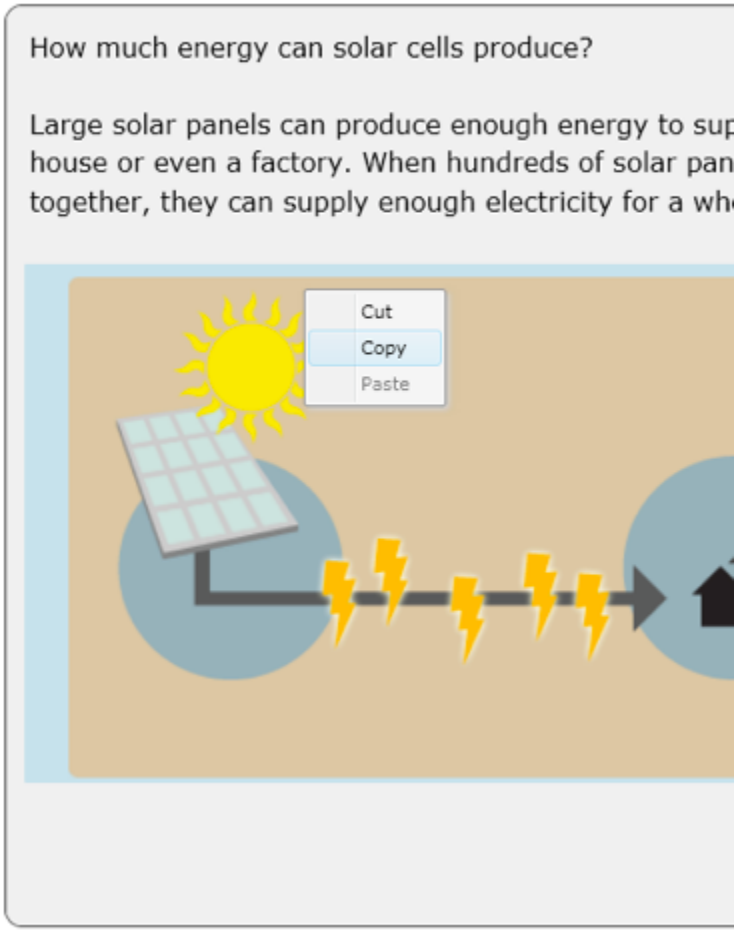
PowerPoint



Composer

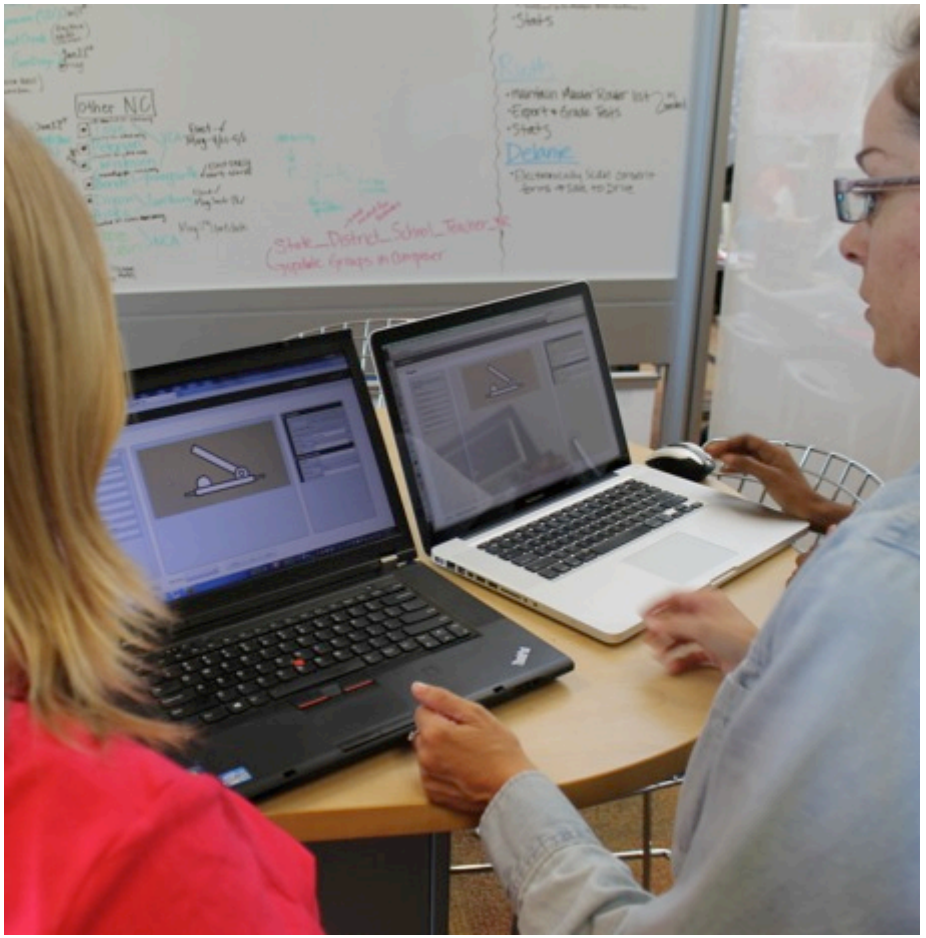
Principle 2: Standard Editing Features

- Relied upon by authors
  - Copy, Cut, and Paste
  - Undo and Redo
  - Revision Tracking
- Can profoundly affect curricular content storage
- Should not be left as a feature to be added at the end of project



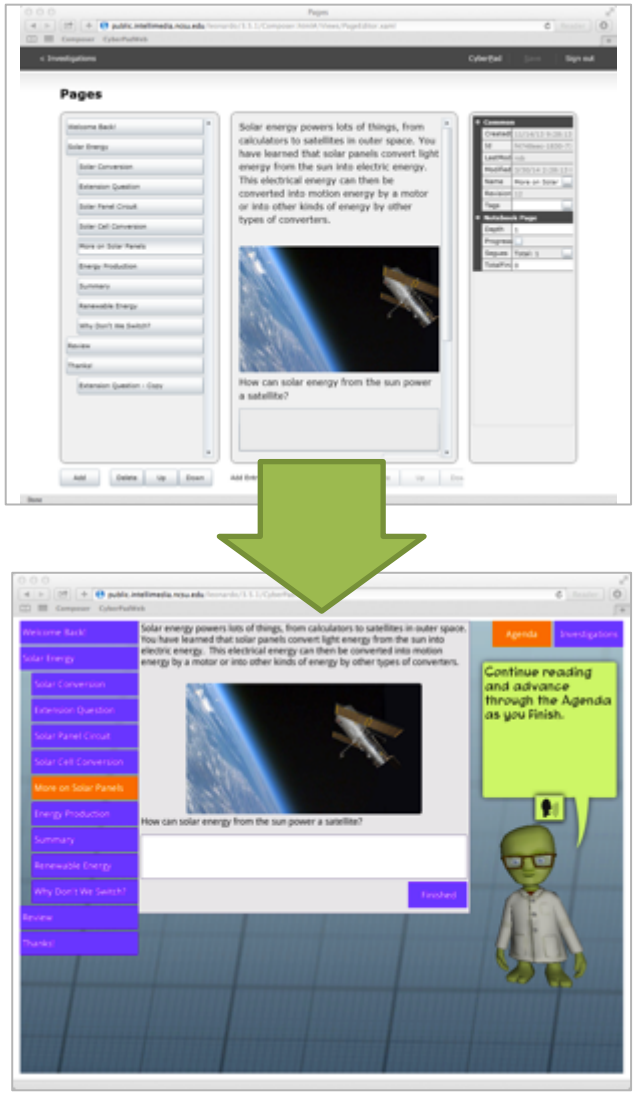
Principle 3: Author Collaboration

- Multiple author collaboration
- Facilitates communities of authors
- Increase both quality and quantity of content



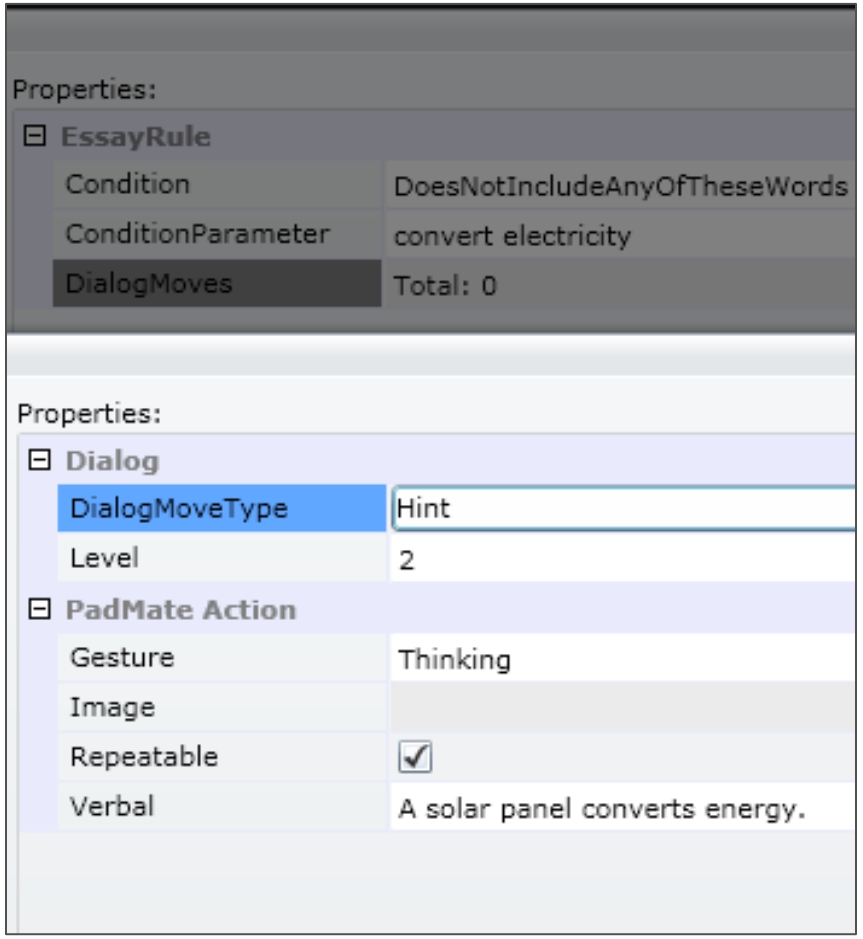
Principle 4: Rapid Iteration

- WYSIWYG or live connection to the ITS
- Changes can be quickly seen in the context of the ITS
- Test ITS behavior while editing rules



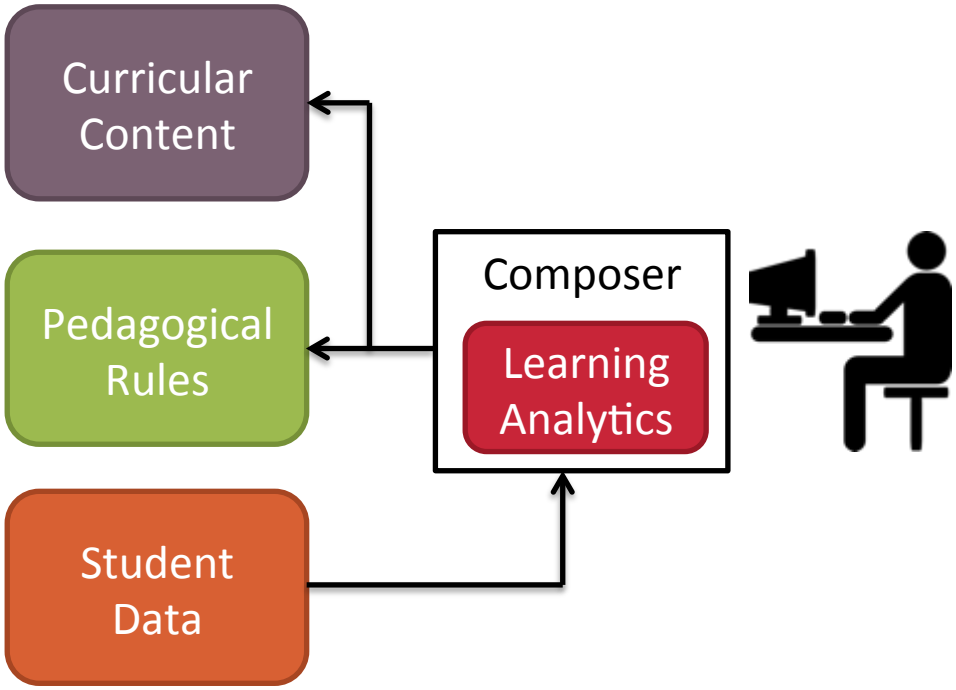
Principle 5: Accommodate Novice and Expert Authors

- UI tailored to novice and expert users
- Wizard interfaces for novice users
- Advanced authoring UI for expert users



Principle 6: Automation

- Some tasks too labor intensive
- Provide automation for repetitive tasks
- Learning analytics to highlight curricular "hot spots"



Outline

- Related Work
- Case Study: Authoring the LEONARDO Digital Science Notebook
- Design Principles for ITS Authoring Tools
- Conclusion and Future Work

Conclusion

- Widespread deployment of ITSs depends on efficient transfer of domain knowledge
- Authoring tool design can leverage decades of software UI and workflow evolution
- Authoring tools hold great promise to facilitate rapid creation of ITSs

Future Work

- Apply learning analytics to identify parts of the curriculum that need additional scaffolding
- Introduce intelligent user interface to facilitate accelerated development of content by novice authors
- Investigate the effects of these capabilities on author efficiency and curricular content quality

Acknowledgements

Collaborators

- John Bedward, *STEM Education*
- Courtney Behrle, *STEM Education*
- Michael Carter, *English*
- Kirby Culbertson, *Art and Design*
- Christopher Mitchell, *Computer Science*
- Lindsay Patterson, *STEM Education*
- Jonathan Rowe, *Computer Science*
- Marc Russo, *Art and Design*
- Wayne Sheffield, *STEM Education*
- Angela Shelton, *STEM Education*
- Ruth Sirkin, *STEM Education*
- Eric Wiebe, *STEM Education*

Support

National Science Foundation:

DRL-1020229

Contact

rgtaylor@ncsu.edu