



THE UNIVERSITY OF AUCKLAND  
[www.auckland.ac.nz](http://www.auckland.ac.nz)

# Reciprocity: an Integrated, Continuous Approach to Software Training Authoring, Delivery and Monitoring

April Nixon and John Grundy  
University of Auckland  
New Zealand



# Outline

- ❖ Motivation
- ❖ Overview of our approach
- ❖ Example usage of Reciprocity
- ❖ Architecture and implementation
- ❖ Evaluation
- ❖ Conclusions and Future Research

# Motivation

- ❖ Learning curve for new software systems
- ❖ Software documentation approaches:
  - Hard-copy manual, PDF files
  - HTML help, context-sensitive help
  - Documentation generation tools
- ❖ Computer-supported instruction approaches:
  - Active tutorials, computer-assisted learning by play-back
  - Intelligent tutoring, animations
  - On-line tutorials/help/support

# Example: Orion Systems Ltd Rhapsody Message Mapping tool

The screenshot displays the Symphonia Mapper interface. The main window shows a mapping configuration for `map mainPatientMessage( [in] PatientMessage in, [out] PVisitMessage out )`. The input side includes `PatientRecord` and `PatientNameRecord`, with sub-elements like `Lname`, `Fname`, `PatientID`, `PatientAddress`, `PatientVisits`, `VisitRecord`, `Visit`, and `TreatmentRecord`. The output side includes `PIDField` and `MedRecNumField`.

A detailed view of the `map MapTreatmentRecordToTreatRecord( [in] TreatmentRecord In, [out] TreatRecord Out )` mapping is shown in the foreground. It lists the following fields:

- In:** TreatmentCodeField, TreatmentTypeField, TreatmentDateRecord, TreatmentUnitsField, TreatmentQuantityField, TreatmentCostField
- Out:** TreatDateField, TreatKindField, CostingField, CostField, TaxField

The code editor at the bottom shows the following mapping logic:

```

map MapTreatmentRecordToTreatRecord( <- Input::TreatmentRecord In, -> Output::TreatRecord Out )
{
    MapDateRecordToString( In.TreatmentDateRecord.DateRecord, Out.TreatDateField.#PCDATA );
    Out.TreatKindField.#PCDATA = In.TreatmentTypeField.#PCDATA;
    Out.TaxField.#PCDATA = In.TreatmentCostField.#PCDATA * 0.1;
    Out.CostField.#PCDATA = In.TreatmentCostField.#PCDATA * 0.9;
    if(In.TreatmentQuantityField.#PCDATA > 1)
    {
        Out.CostingField.#PCDATA = "costing1";
    }
    else
    {
        Out.CostingField.#PCDATA = "costing2";
    }
}

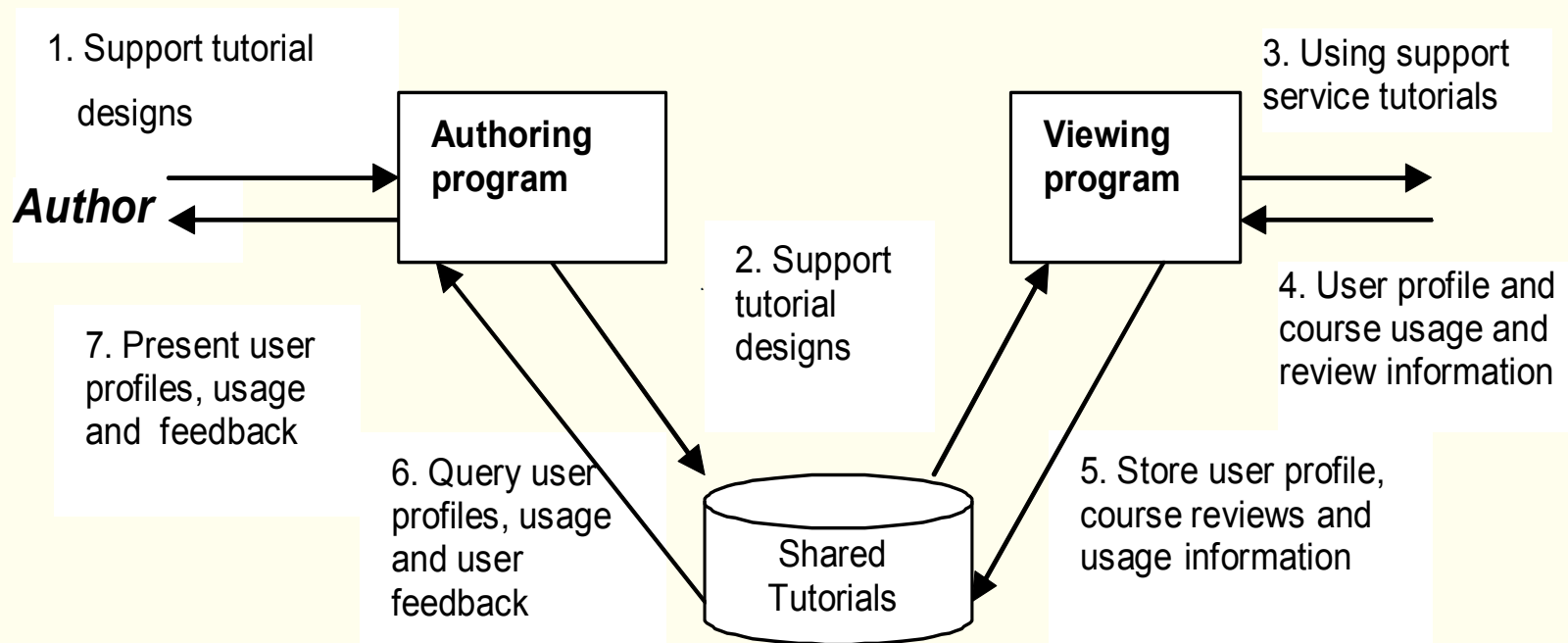
```

The interface also shows a message log at the bottom left indicating "Mapping File... Mapping Complete." and an "Output" window at the bottom right.

# Issues in Documenting

- ❖ Orion's documentation team builds PDF tutorials
- ❖ Interaction with clients difficult
- ❖ Distributing tutorials & updates difficult
- ❖ Feedback from users virtually impossible
- ❖ Can't tailor tutorials to different user group needs
- ❖ PDF very "static" approach to tutorial
- ❖ Integration with Message Mapper poor

# Reciprocity Approach



# Tutorial Authoring

- ❖ Concept of lessons:
  - Target application tutorials
  - Tutorial has set of Courses
  - Each course has set of Lessons
  - Lesson index gives set of Subjects
  - Subjects made up of one or more Pages – text, graphics, animations, user interaction
- ❖ Pages/subjects/lessons/tutorials can be rated
- ❖ Different users have access to different lessons etc.

# Tutorial Authoring

The screenshot displays the Reciprocity Author software interface. The main window shows a course structure on the left, a central text editor with an 'Installation' page, and an 'Action Editor' window for an 'InstallShield Wizard'. A 'Rating Page Specification' dialog box is open, showing criteria for a rating page. The dialog includes a table with columns for 'Title' and 'Choices', and buttons for 'Add Criteria', 'Update', 'Delete', 'Cancel', and 'OK'. A 'Single Click Parameters' dialog box is also visible, showing coordinates and a list of actions like 'single click', 'double click', etc.

**Rating Page Specification**

Title	Choices
Helpfulness	[Not helpful][Slightly helpful][Helpful][Very Hel...
Overall	[Poor, 1][2][3][4][Excellent, 5]

ask for comments

Buttons: Add Criteria, Update, Delete, Cancel, OK

**Single Click Parameters**

Action Region

x coordinate: 329

y coordinate: 343

height: 21

Instructions:

1. After reading the instructions on the screen, click once with the left mouse button on the Next button to progress to the next step in the dialog.



# Lesson Usage

The screenshot shows a web browser window titled 'Reciprocity Viewer' displaying a course list for 'April Nixon'. The selected course is 'Symphonia Mapper: Getting Started'. The interface includes a navigation menu on the left with sections like 'Background and Installation', 'Using the Symphonia Mapper', and 'Main Maps'. The main content area shows an 'Installation' section with an 'InstallShield Wizard' window overlaid. The wizard window displays a blue graphic and the text: 'Welcome to the InstallShield Wizard for Symphonia Mapper. The InstallShield® Wizard will install S... your computer. To continue, click Ne...'. At the bottom of the wizard are buttons for 'First', 'Previous', 'Next', 'Last', and 'Try'. A 'Ready' status bar is visible at the bottom left.

Overlaid on the right is an 'Inbox' window showing a list of emails:

Unread	Sender	Subject	Date
<input type="checkbox"/>	April Nixon	Time test message	25 Sep, 2001 3:11:...
<input type="checkbox"/>	April Nixon	Re: Test message	1 Oct, 2001 1:29:0...
<input type="checkbox"/>	April Nixon	Re: Re: Test message	1 Oct, 2001 1:55:2...
<input type="checkbox"/>	April Nixon	Test at 2:20	1 Oct, 2001 2:21:0...
<input type="checkbox"/>	John Grundy	Help needed re Lesson 1: Opening Docu...	1 Oct, 2001 5:07:4...

Below the inbox is a 'Course Ratings: Symphonia Mapper: Getting Started' window. It features a rating scale from 1 to 5, with the 5th option selected. Below the scale is a text area for comments, with the text 'Hi th... I hav... Lesso... Thank... End U...' visible. A 'Submit' button is located at the bottom right of the rating window.

# Feedback to Authors

The screenshot displays the Reciprocity Author software interface. The main window shows a course tree on the left and a details pane on the right. The details pane includes sections for General Details, Software Details, Support Package Details, and Summary Statistics. Two dialog boxes are overlaid on the main window:

**New Message Dialog:**

- Recipient(s): <All users>
- Subject: Updated
- Message body: To all users, The second lesson in this course includes two new subjects. It also includes customised shortcuts and examples.
- Signature: Course Author, April Nixon
- Buttons: Cancel, Send

**Rating Criteria Specification Dialog:**

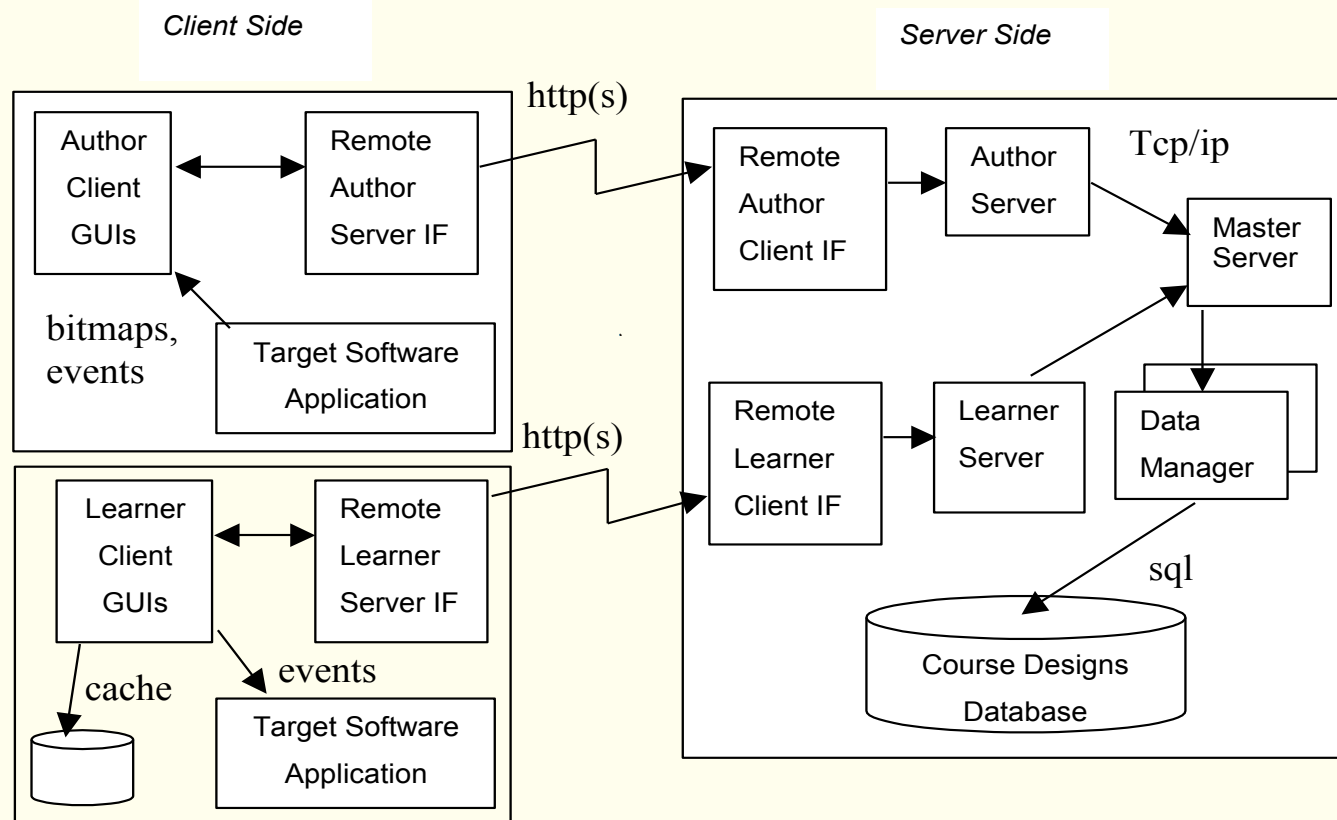
- Criteria Title: Overall
- Choices table:

Category	Value
Poor	1
	2
	3
	4
Excellent	5

New Choice section:

- Category:
- Value:
- Buttons: Add, Delete, Cancel, OK

# Tool Architecture



# Evaluations

## ❖ Two user surveys:

- Authors (Orion staff)
- Users (Orion staff, Orion users, students)
- Authors found most organisation & feedback facilities to be good, but desired greater authoring support and ability for users to do “tests” at end of lessons
- Users found tutorial navigation, tailoring and content to be good, but desired greater tool animation and help facilities
- Prototype only deployed with small number of authors/users so unclear how scalable; issues with content creation support; issues with caching lessons etc.

## Evaluation #2

### ❖ Two qualitative:

- Using CAI assessment criteria
- Using general usability criteria
- Supports integrated learning and usability well: encourages active users, appropriate use of multi-media, simple navigation and supporting learner feedback to authors
- Areas for improvement navigation support, allow learners to “try again” during animated demonstrations, provision of a testing facility for learners to gauge progress
- Concurs with Orion staff assessment from user survey

# Future Research

- ❖ Allow third-party authoring tools to provide content for tutorials
- ❖ Better integration with target applications
- ❖ Better animation control of target applications
- ❖ Navigation/search improvement essential
- ❖ Need organisational support for active tutorial content development and delivery

# Conclusions

- ❖ Learning to use software applications is hard – nothing likely to ever fully solve this...
- ❖ Various approaches exist to support learning, but author/user feedback cycle very limited or non-existent
- ❖ Reciprocity demonstrates that it is feasible to provide this feedback loop for software documentation/tutorial authoring and delivery

# Acknowledgements

- ❖ Orion Systems Ltd
- ❖ Dave Brewerton
- ❖ Mark Capill
- ❖ Technology New Zealand