



# Adaptive Tailoring of Student Learning: Notes on a Soar Approach

25 Jun 2009

Brian Stensrud, Bob Wray, Laura Hamel  
Soar Technology, Inc.  
{stensrud, wray, lhamel}@soartech.com

# Key Limitation of Game-based Practice Environments

- Game-based practice environments provide a trainee with a realistic synthetic environment in which he can practice some particular skill or set of skills
- Practice environments alone are insufficient to promote learning
  - When the trainee gets lost/stuck/confused, he has no support
- Practice environments need “pedagogical experience management” (Lane and Johnson)
  - Configuration of scenarios/environment focused on student needs
  - Coaching via individualized guidance and feedback
  - **Tailoring of training experience to individualized needs**
  - Individualized/contextual after-action review

## Goal: General Dynamic Tailoring Technologies

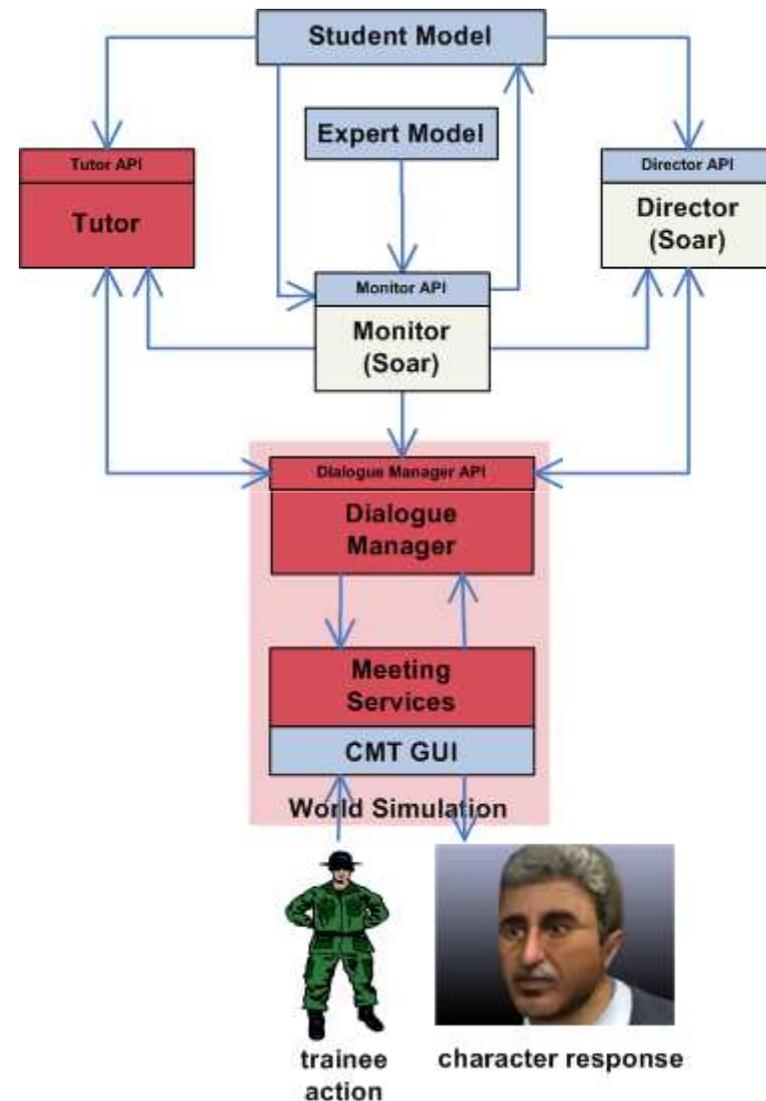
- Relative to configuration, coaching/tutoring, and explanation/AAR dynamic tailoring is relatively unexplored
- Key Capability: Support individualized, targeted, dynamic delivery of guidance & feedback in game-based practice environments
  - Informed by deliberate instructional strategies
  - Tailored to individual student needs
  - **Enabled by robust student model(s)**

# What is a student model?

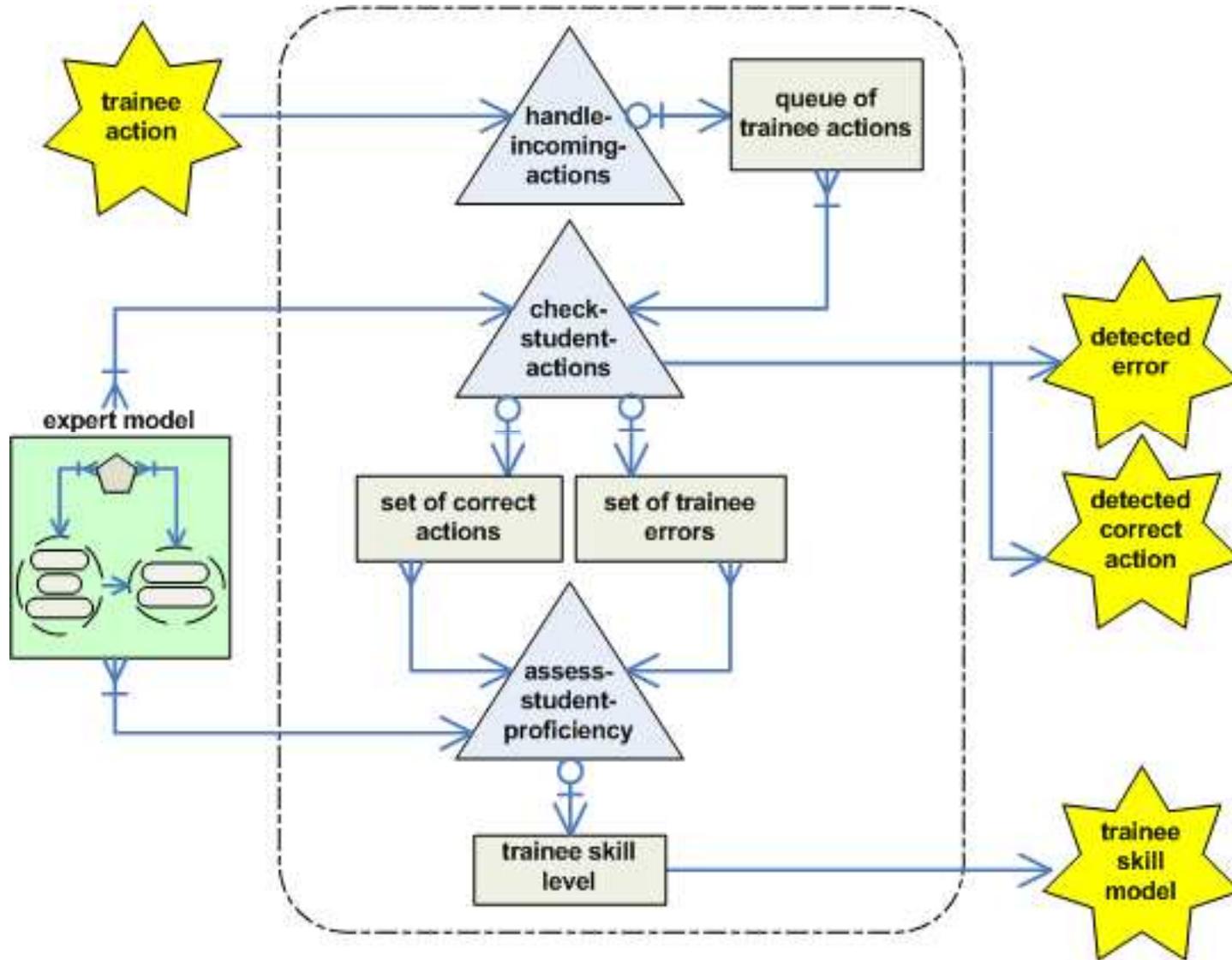
- The component of a training system that contains estimates about a student's knowledge and skill
  - Dynamic assessment, updated with every student action
  - Can be used to provide summative assessment information
  - Can be used as the basis for providing individualized instruction
- The degree of individualization is driven by the model's representation and content
  - Guidance
  - Feedback on performance (real-time and post-instruction)
  - Variations in instructional content

# Summary of Current Approach

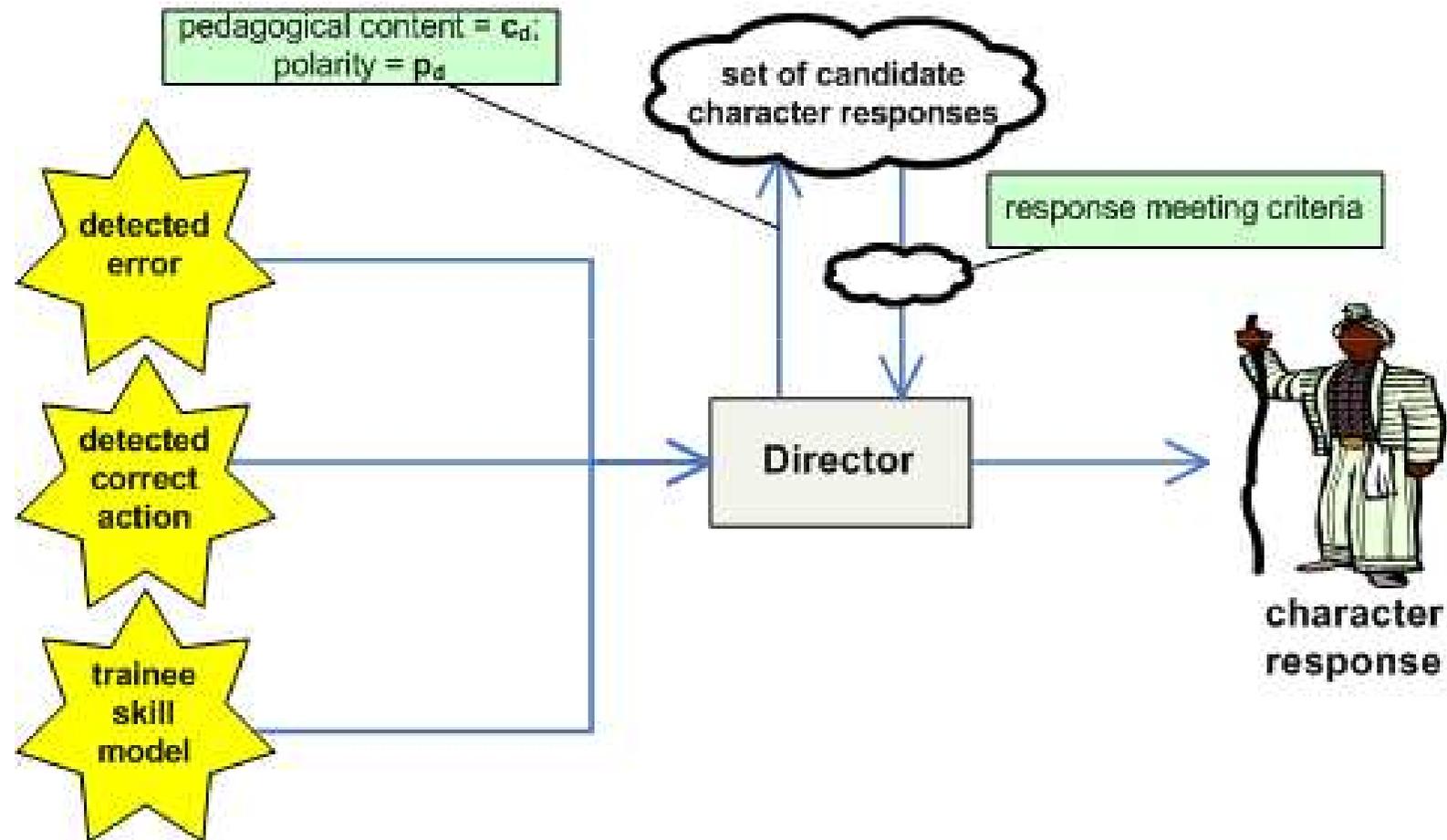
- Domain: two-person dialog
- Declarative Models of Expert/Student Knowledge & Errors
  - Expert Model implicitly defines correct behavior via constraint definitions
  - Sufficient to model expertise and skill in an ill-defined domain
  - Enables dynamic assessment of student actions
  - Guide pedagogical experience management during practice
- Monitor interprets student action
  - Primary process for monitoring student activity; updating models
  - Suggest actions for coach and director
  - Key role of the monitor is to read and update information in the student models
- Director chooses tailoring responses based on Monitor's assessment
  - Character response content
  - mannerisms, tone, facial expressions



# Soar Implementation of Monitor



# Soar Implementation of Director - Character Response Selection



# Tailoring experience to provide guidance/feedback

- Target tailoring capabilities:
  - More independent practice environment, requiring students to rely more heavily on realistic cues from the environment
  - More structured experience than free-play

<b>Tailoring strategy</b>	<b>Example</b>
<b>Control of available actions</b>	Novice students only have access to phase-relevant actions to help them learn sequencing. Other actions added in as student gains experience and/or expertise.
<b>Delivery of feedback through game characters</b>	Student's negotiation partner may suggest a transition to the next meeting phase.
<b>Controlling consequences of errors</b>	Minimize negative impacts of incorrect actions to make it possible for novices to advance in the narrative despite
<b>Simplifying environment</b>	Control randomization factor so novices aren't confused by negative responses to their correct actions.

# Nuggets and Coal



Soar provides a lot of flexibility making decisions based on the current situation



Declarative models + general monitoring capability enables rapid changes to content



Dynamic tailoring represents a functional challenge goal where many of Soar's strengths are evident



SML/ATE XML translation makes working with XML structures easy



Not taking advantage of all Soar could provide (specifically, could benefit from learning strategies).



Monitor implementation not really as domain-independent as it could be



Soar is an awkward fit with practice environment, implemented in Java... JSoar?

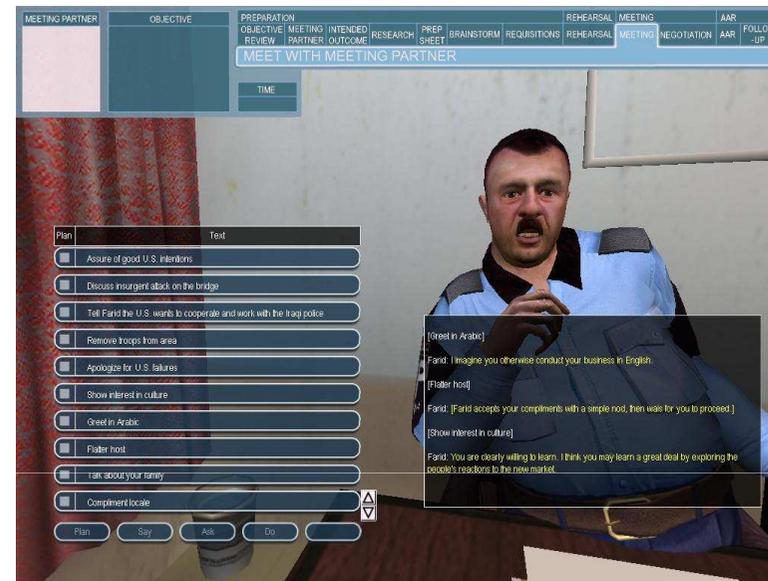


Explicit representation of expert model (versus constraint-based) may be preferable?

# Backup Slides

# Current Domain Focus: Intercultural Meetings

- Domain is a subset of ICT's ELECT BiLAT Training Game
  - Cultural training game implemented in Unreal game engine
  - Developed by ICT/CCT at USC
- ELECT BiLAT is focused on bilateral engagement
  - Negotiation activities aimed at bringing about agreement
  - Social and cultural considerations in negotiation



- Primary game phases
  - Meeting preparation
  - Meeting rehearsal
  - **Meeting conduct**
    - Training objectives
      - Establish rapport; small talk
      - Awareness of cultural cues
  - Meeting AAR



# Research Testbed: Cultural Meeting Trainer

## Prototype

- Backend elements adapted from ELECT BiLAT implementation
- Browser-based, GUI front-end (Flash)
- Includes previously designed instructional content and coaching capability

## Focus to-date

- Achieve functionality sufficient for tailoring & coaching explorations

## Potential Future enhancements

- Improve playability design & graphical content
- Audio and alternative input options
- UI Adaptations

