

Human Language Learning as a Metaphor for Teaching CS1-2

Edward L. Jones
Florida A&M University



8 November 2010

1



Outline

- Challenges of Teaching CS1-CS2
- Trends – Student Engagement
- Sage Advice Revisited
- Human Language Learning Metaphor
- An Example
- What Next?
- Acknowledgments

8 November 2010

2

- Negative indicators
 - High failure rates (up to 65%)
 - Poor mastery
 - Undesired behavior -- plagiarism
- Under the hood ...
 - Lecture based
 - Program-construction dominant activity
 - Individual mastery vs collaboration
- An even closer look
 - Cognitive jungle (Bloom's taxonomy)
 - So many "languages" (skills) to learn

- FUN!! – robotics, gaming, multimedia ...
- Active Learning
 - Increased interaction, facilitated learning
 - Reduced role of lectures
- Studio-Based Learning
 - Collaborative problem solving
 - Feedback / reflection
- Costs
 - Technology support
 - Behavior changes

- **MORE** of this:
 - Study examples of good software
 - Exercises to modify / combine programs
 - Experience using existing software solutions
- **LESS** of this:
 - Programming from scratch
 - Assignment of throw-away programs
 - Emphasis on abstract skills at the expense of concrete experience

It's not the advice you'd expect. Learning a new language seems formidable as we recall from years of combat with grammar and translations in school. Yet infants begin at birth. They communicate at eighteen months and speak the language fluently before they go to school. And they never battle translations or grammar explanations along the way.

Then into a world of language jargon, children figure out language purely from the jargon, objects and interactions around them.

That's access for optimal circuits that send the stimuli to different language areas in the brain. Meanings flow to words. Words string into structures. And language erupts.

Three characteristics of the child's language-learning process are crucial for success:

First, and most importantly, a child's natural language learning ability emerges only in a speech-saturated, immersion environment free of translations and explanations of grammar.

Second, a child's language learning is dramatically accelerated by constant feedback from family and friends. Positive correction and persistent reinforcement nurture the child's language and language skills into full communicative expression.

Third, children learn through play, whether it's the utter writing following act that announces their first step or the giggling prearrange to their first words. All the conversational chatter skittering through young children's play with parents and playmates—"what's this?"—"clap, clap your hands."—"my ball..."—helps children develop language skills that connect them to the world.

Adults possess this same powerful language learning ability that we've lost over the years. Sadly, our classes with vocabulary drills and grammar explanations force us to conclude it's hopeless. We simply don't have "the language learning gene."

At Rosetta Stone® we know otherwise. You can recover your native language learning ability as an adult by prompting your brain to learn language the way it's wired to learn language: by

What's the fastest way to learn a language?

ACT LIKE A BABY.

complete immersion. Our award-winning, computer-based method does just that.

Dynamic Immersion® unlocks the innate language-learning ability you acquired before birth and mastered as a child.

By recreating the immersion context in which you learned your first language, you understand, speak, read and write your new language with confidence and accuracy from the beginning—without translations and explanations.

Acquire step-by-step **listening, speaking, reading and writing** skills, plus **grammar, syntax, idiomatic feedback**, including speech recognition and analysis technologies that prepare you for everyday conversations. And Adaptive Recall™ brings back material just when you need it to reinforce and perfect your learning.

Every act of learning is an act of play for children and there's no reason it should be different for learners of any age. With Rosetta Stone programs, you rediscover the joy of learning language. Cliven, puzzle-like activities produce sudden "Aha!" moments and astonishing language discoveries.

Your "language brain" remembers. We see it all the time.

A slow smile spreads across the learner's face after just a few screens. It's a smile of recognition, as though the brain suddenly recalls what it was like to learn language as a child, as though it recalls, "Aha! I've done this before!"

Act like a baby!™ This isn't just how you can maximize your own innate language learning ability with Rosetta Stone. It's the fastest way to learn a language. Guaranteed.™

More than 20 languages available.

Level 1	Prog.	\$199	NEW \$154
PREPWORK			
Level 1&2	Prog.	\$499	NEW \$349
PREWORK			
Level 1&2	Prog.	\$1,099	NEW \$849
PREWORK			

SAVE 10% plus FREE SHIPPING

Three characteristics of the child's language-learning process are crucial for success:

First, and most importantly, a child's natural language-learning ability emerges only in a speech-soaked, immersion environment free of translations and explanations of grammar.

Immersion

Second, a child's language learning is dramatically accelerated by constant feedback from family and friends. Positive correction and persistent reinforcement nurture the child's language and language skills into full communicative expression.

Feedback

Third, children learn through play, whether it's the arm-waving balancing act that announces their first step or the spluttering preamble to their first words. All the conversational chatter skittering through young children's play with parents and playmates—"...what's this..." "...clap, clap your hands..." "...my ball..."—helps children develop language skills that connect them to the world.

Play, Fun!

- Language(s) learned via immersions
 - Speaking → Reading → Writing
 - Social interaction is key (AL/SBL)
 - Outcomes include critical thinking and communication
- Software is multi-lingual
 - Requirements, specification, design, code, application
 - Code is just one of the many languages
- HLLM → Macro design of holistic learning experiences

Sample Holistic Assignment

Charter	Understand intent; Code from design.
Inputs:	(1) Need statement; (2) Flowchart; (2) Solution key (executable only);
Solution stages:	
WHAT	Create problem instances from Need; Execute solution key to verify instances; Critique/discuss discrepancies.
HOW	Goal: understand/verify design. Trace problem instances thru design. Critique/discuss discrepancies.

An Example (cont'd)

BUILD	Code the solution design; Verify code compliance to design; Reflect/discuss required corrections.
VERIFY	Execute code on problem instances. Reflect/discuss discrepancies in code design, need statement.
SUBMIT	Goal: Retain artifacts for future use. Submit <u>all</u> artifacts.

- #0: Programming is multilingual
- #1: Consume before producing
- #2: Drill for skill
- #3: No answer is ever really wrong –
Teachable moments (reuse artifacts)
- #4: Invest in infrastructure

- Technical specifications varying in formality of syntax and precision of semantics
 - Problem specification (intent)
 - Design specification (guidance)
 - Code (unforgiving syntax, semantics)
- Translation across specifications

- Ability to read a language should precede a demand to write in it
- Higher-order language constrains; lower-order language elaborates
 - Write code based on a design
 - Given a specification, critique design
 - Given requirements, critique candidate solutions

- Mastery of basic skills a prerequisite to participating in active learning
- Lecture no substitute for dirty hands!
- Require coding/debugging drills on selected construct(s)
- Drills must not be optional
 - Count significantly in grade
 - Instrumented drill environment

- Making mistakes is inherent in programming
- Recognizing and fixing mistakes develops critical thinking skills and provides opportunity for interactive group learning
- Students refine/verbalize their “value systems” for good/correct software

- **INERTIA:** No extra effort/resources needed for “business as usual”
- Significant infrastructure needed to support HLLM
 - Multilingual software artifacts
 - Instrumented drill environments
 - Observational environment (behavior, intervention)
 - Holistic assignment templates
 - Army of TAs!!

- Create HLLM lesson plan templates
- Create artifacts repository
- Acquire/develop tools for drills
- Elaborate lesson plan templates as individual & group assignments
- Assess costs and impacts

- This work is supported by NSF CPATH Award CNS-0939138 – CT-Driven Evolution to Studio-Based Active Learning

Questions??