

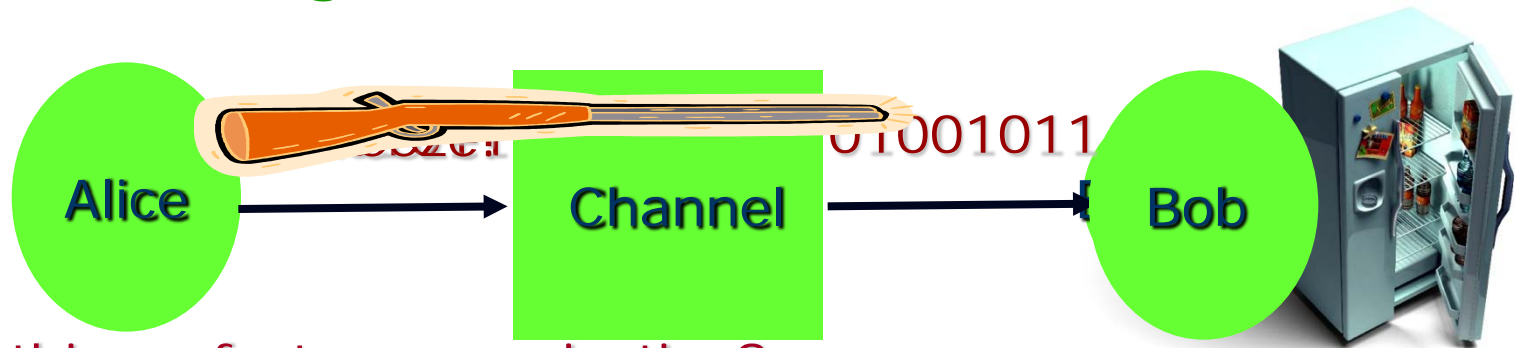
# Universal Semantic Communication

**Madhu Sudan**

Microsoft Research + MIT

Joint with **Oded Goldreich** (Weizmann) and **Brendan Juba** (MIT).

# The Meaning of Bits



- Is this perfect communication?
- What if Alice is trying to send instructions?
  - Aka, an algorithm
  - Does Bob understand the correct algorithm?
  - What if Alice and Bob speak in different (programming) languages?

# Part I: Context/Motivation

# What? Why?

- Example 1: I have a presentation that used to work well on my last laptop.

- Distance:  $\delta(f, g) = \Pr_{x \in D}[f(x) \neq g(x)]$   
 $\delta(f, \mathcal{F}) = \min_{g \in \mathcal{F}} \{\delta(f, g)\}$   
 $f \approx_{\epsilon} g$  if  $\delta(f, g) \leq \epsilon$ .
- Definition:  
 $\mathcal{F}$  is  $(q, \alpha)$ -locally testable if

- I transferred the file to my new laptop and it looks like this.

- Distance:  $\pm(f, g) = \Pr_{x \in D}[f(x) \neq g(x)]$   
 $\pm(f, \mathcal{F}) = \min_{g \in \mathcal{F}} \{\pm(f, g)\}$   
 $f \approx_{\epsilon} g$  if  $\pm(f, g) \leq \epsilon$ .
- Definition:  
 $\mathcal{F}$  is  $(q, \alpha)$ -locally testable if

- ... but the bits are intact!

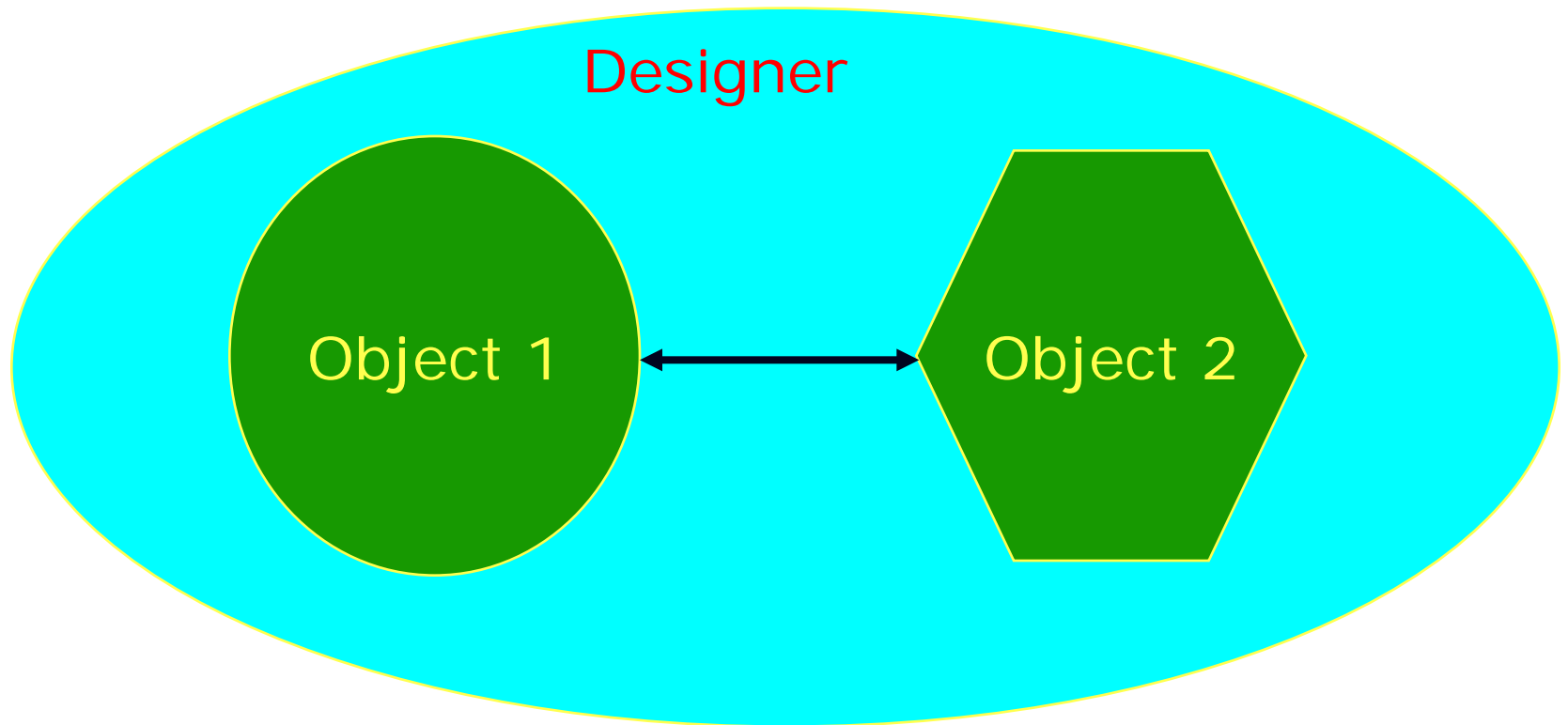
# What? Why?

- Example 2: I would like to print some document on some printer.
  - You can do it.
  - I have same permissions as you.
  - But I don't have the printer installed.
- I have the information ... I don't know how to translate to printer's language.

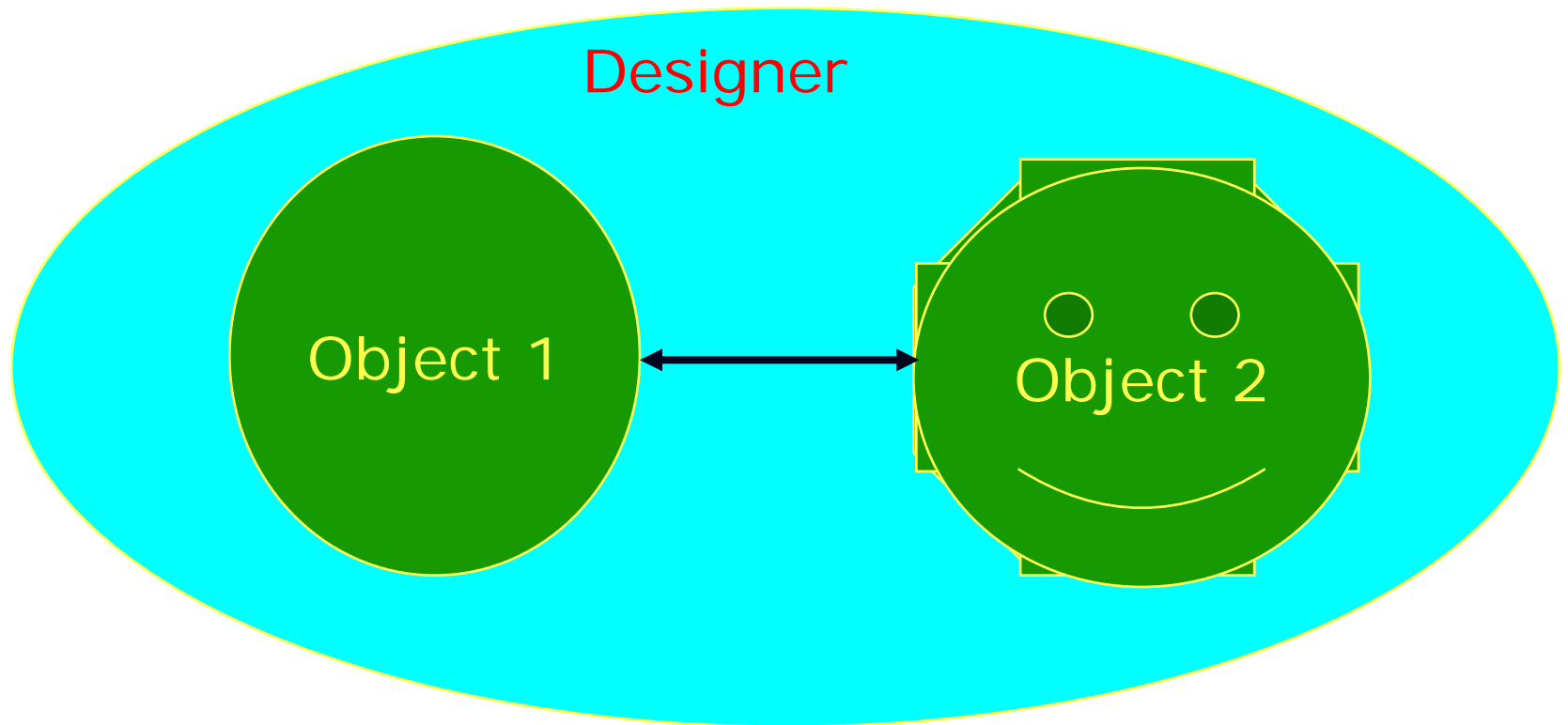
# Motivation: Better Computing

- Computers are constantly communicating.
- Networked computers use common languages:
  - Interaction between computers (getting your computer onto internet).
  - Interaction between pieces of software.
  - Interaction between software, data and devices.
- Getting two computing environments to “talk” to each other is getting problematic:
  - time consuming, unreliable, insecure.
- Can we communicate more like humans do?

# Classical Paradigm for interaction



# New paradigm





# Bits vs. their meaning

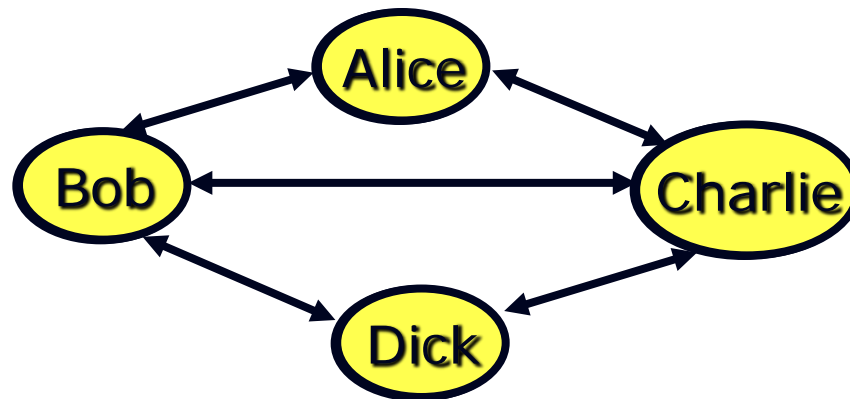
- Say, Alice and Bob know different programming languages. Alice wishes to send an algorithm  $A$  to Bob.
  - $A$  = sequence of bits ... (relative to prog. language)
- **Bad News:** Can't be done
  - For every Bob, there exist algorithms  $A$  and  $A'$ , and Alices, Alice and Alice', such that Alice sending  $A$  is indistinguishable (to Bob) from Alice' sending  $A'$
- **Good News:** Need not be done.
  - From Bob's perspective, if  $A$  and  $A'$  are indistinguishable, then they are equally useful to him.
- What should be communicated? Why?

## Aside: Why communicate?

- Classical "Theory of Computing"



- Issues: Time/Space on DFA? Turing machines?
- Modern theory:



- Issues: Reliability, Security, Privacy, Agreement?
- If communication is so problematic, then why not "Not do it"?

# Motivations for Communication

- Communicating is painful. There must be some compensating gain.
- What is Bob's Goal?
  - "Control": Wants to alter the state of the environment.
  - "Intellectual": Wants to glean knowledge (about universe/environment).
- Claim: By studying the goals, can enable Bob to overcome linguistic differences (and achieve goal).

# **Part II: Computational Motivation**

# Computational Goal for Bob

- Why does Bob want to learn algorithm?
  - Presumably to compute some function  $f$   
(A is expected to compute this function.)
  - Lets focus on the function  $f$ .
- Setting:
  - Bob is prob. poly time bounded.
  - Alice is computationally unbounded, does not speak same language as Bob, but is "helpful".
  - What kind of functions  $f$ ?
    - E.g., uncomputable, PSPACE, NP, P?

# Setup

~~Bob~~ User

$f(x) = 0/1?$

$R \leftarrow \$ \$ \$$


~~Alice~~ Server

$q_1$



Different from interactions in  
cryptography/security:

There, User does not **trust** Server,  
while here he does not  
**understand** her.



Computes  $P(x, R, a_1, \dots, a_k)$

Hopefully  $P(x, \dots) = f(x)$ !

# Intelligence & Cooperation?

- For User to have a non-trivial interaction, Server must be:
  - Intelligent: Capable of computing  $f(x)$ .
  - Cooperative: Must communicate this to User.
- Formally:
  - Server  $S$  is helpful (for  $f$ ) if
    - $\exists$  some (other) user  $U'$  s.t.
      - $\forall x$ , starting states  $\sigma$  of the server  
 $(U'(x) \leftrightarrow S(\sigma))$  outputs  $f(x)$

# Successful universal communication

- Universality: Universal User  $U$  should be able to talk to any (every) helpful server  $S$  to compute  $f$ .
- Formally:
  - $U$  is  $f$ -universal, if
$$\forall \text{ helpful } S, \forall \sigma, \forall x$$
$$(U(x) \leftrightarrow S(\sigma)) = f(x) \text{ (w.h.p.)}$$
- What happens if  $S$  is not helpful?
  - Paranoid view  $\Rightarrow$  output " $f(x)$ " or "?"
  - Benign view  $\Rightarrow$  Don't care (everyone is helpful)



# Main Theorems [Juba & S. '08]

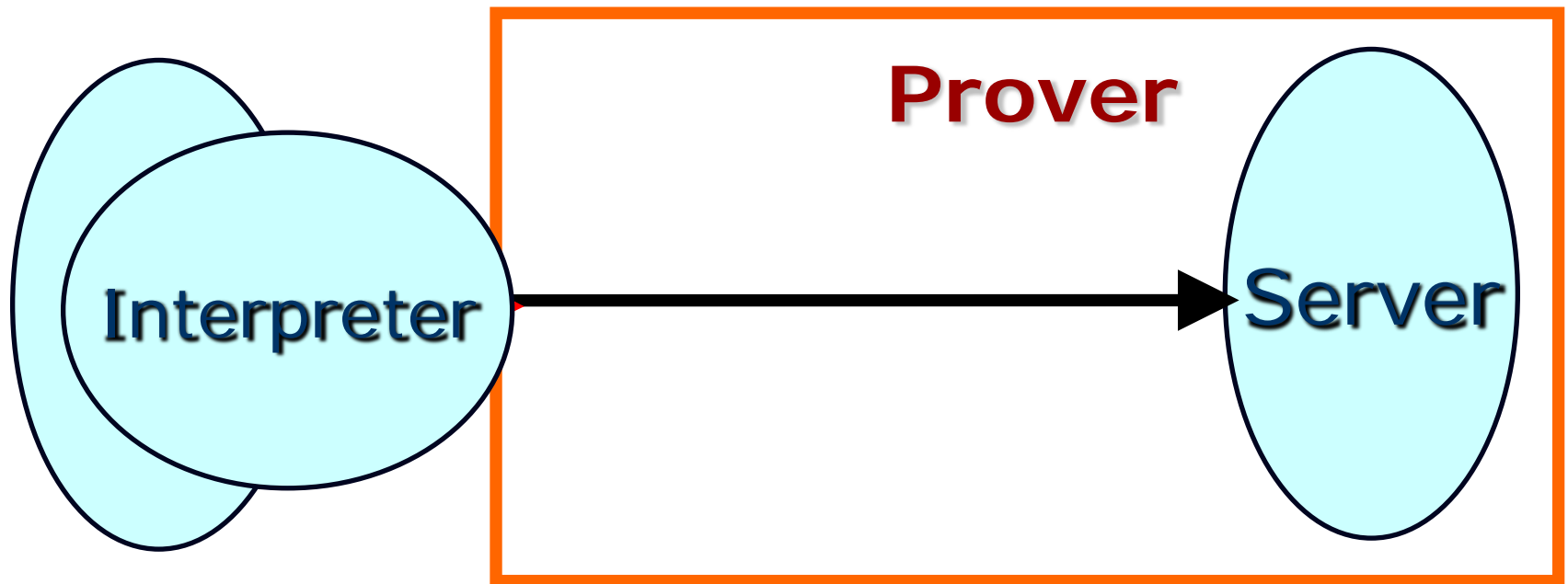
- If  $f$  is PSPACE-complete, then there exists a  $f$ -universal user who runs in probabilistic polynomial time.
  - Extends to checkable problems
    - $(NP \cap co-NP)$ , breaking cryptosystems)
    - $S$  not helpful  $\Rightarrow$  output is safe
- Conversely, if there exists a  $f$ -universal user, then  $f$  is PSPACE-computable.
  - Scope of computation by communication is limited by misunderstanding (alone).

# Implications

- No universal communication protocol ☹
  - If there were, should have been able to solve every problem (not just (PSPACE) computable ones).
- But there is gain in communication:
  - Can solve more complex problems than on one's own, but not every such problem.
- Resolving misunderstanding? Learning Language?
  - Formally **No!** No such guarantee.
  - Functionally **Yes!** If not, how can user solve such hard problems?

## Few words about the proof: Positive result

- Positive result: Enumeration + Interactive Proofs
- Guess: Interpreter;  $b \in \{0,1\}$  (value of  $f(x)$ )



- Proof works  $\Rightarrow f(x) = b$ .
- If it doesn't  $\Rightarrow \{\text{Interpreter or } b\}$  incorrect.

# Proof of Negative Result

- $L$  not in PSPACE  $\Rightarrow$  User makes mistakes.
  - Suppose Server answers every question so as to minimize the conversation length.
    - (Reasonable effect of misunderstanding).
  - Conversation comes to end quickly.
  - User has to decide.
  - Conversation + Decision simulatable in PSPACE (since Server's strategy can be computed in PSPACE).
  - $f$  is not PSPACE-computable  $\Rightarrow$  User wrong.
  - **Warning:** Only leads to finitely many mistakes.

# Principal Criticisms

- Solution is no good.
  - Enumerating interpreters is too slow.
    - Approach distinguishes **right/wrong**; does not solve search problem.
    - Search problem needs new definitions to allow better efficiency.
- Problem is not the right one.
  - Computation is not the goal of communication. Who wants to talk to a PSPACE-complete server?



Next part of talk

# Part III: Generic Goals

# Generic Communcation [Goldreich, J., S.]

- Still has goals. Goals more diverse.
  - Should be studied; defined formally.
- Major types:
  - Control, e.g.
    - Laptop wants to print on printer.
    - Buy something on Amazon.
  - Sensing/Informational:
    - Computing some (hard) function.
    - Learning/Teaching.
    - Coming to this talk.
  - Mix of the two.

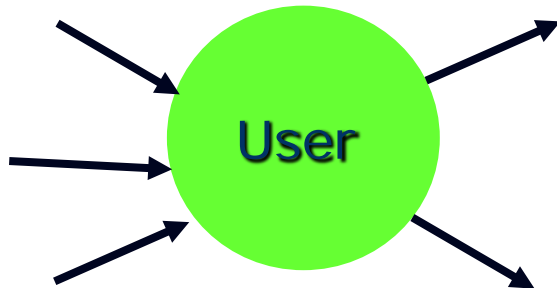
# Universal Semantics in Generic Setting?

- Can we still achieve goal without knowing common language?
  - Seems feasible ...
    - If user can detect whether goal is being achieved (or progress is being made).
  - Just need to define
    - Sensing Progress?
    - Helpful + Universal?
    - ...
    - Goal?
    - User?

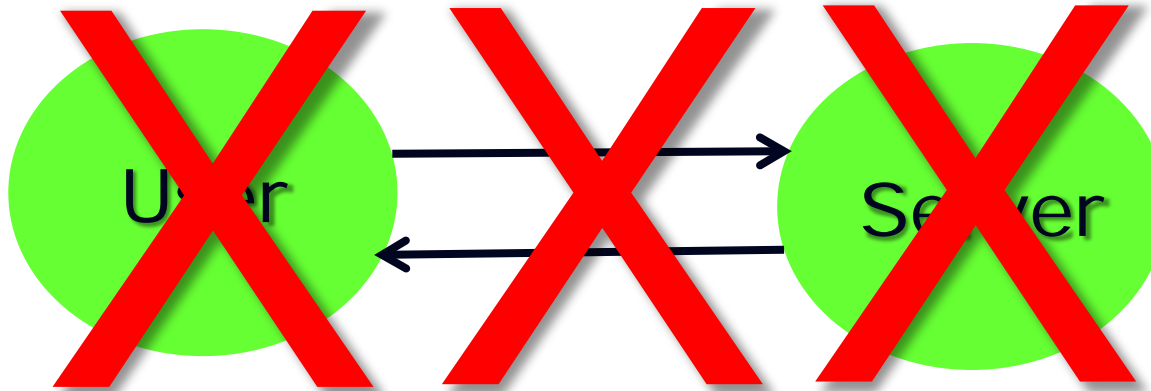


# Modelling User/Interacting agents

- (standard AI model)
- User has state and input/output wires.
  - Defined by the map from current state and input signals to new state and output signals.



# Generic Goal?

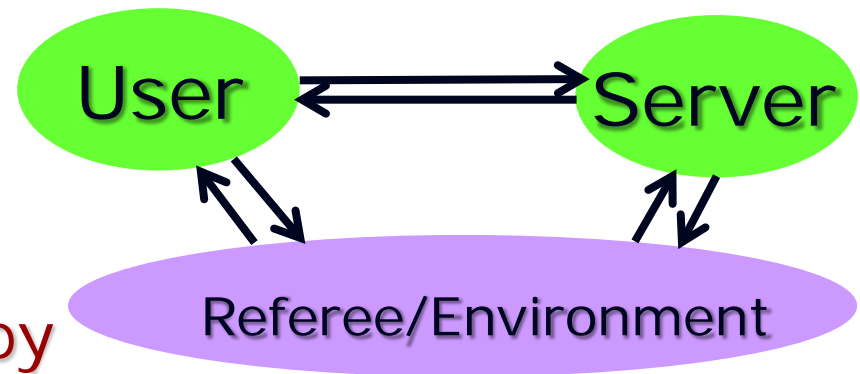


- Goal = function of ?
  - User? – But user wishes to change actions to achieve universality!
  - Server? – But server also may change behaviour to be helpful!
  - Transcript of interaction? – How do we account for the many different languages?

# Generic Goals

- Key Idea: Introduce 3rd entity: Referee

- Poses tasks to user.
- Judges success.

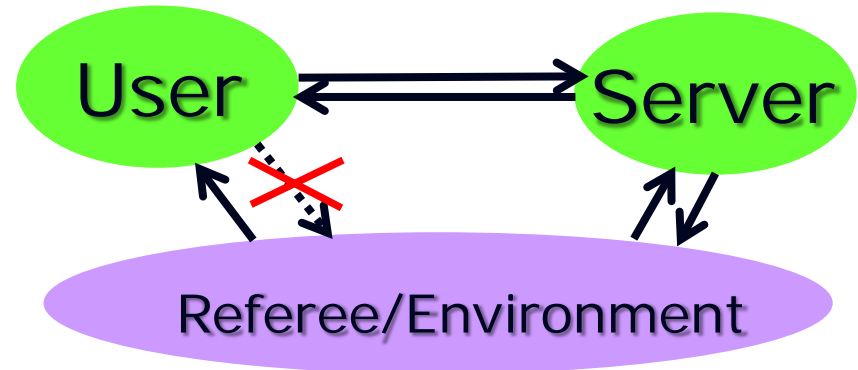


- Generic Goal specified by

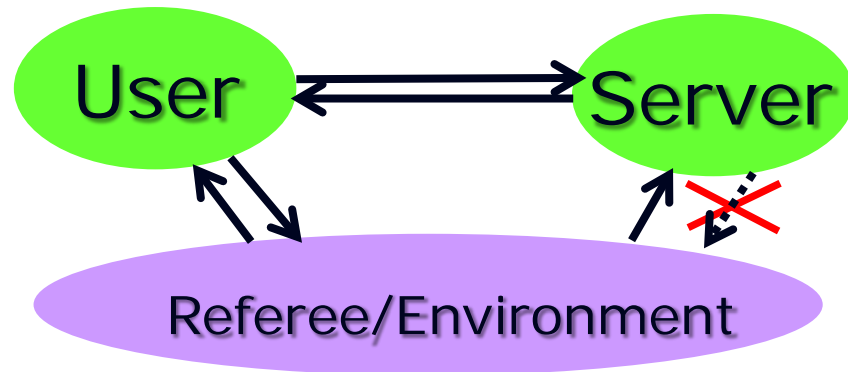
- Referee (just another agent)
- Boolean Function determining if the state evolution of the referee reflects successful achievement of goal.
- Class of users/servers.

# Generic Goals

- Pure Control



- Pure Informational



# Sensing & Universality

- To achieve goal, User should be able to sense progress.
  - I.e., user should be able to compute a function that (possibly with some delay, errors) reflects achievement of goals.
- Generalization of positive result:
  - Generic goals (with technical conditions) universally achievable if  $\exists$  sensing function.
- Generalization of negative result:
  - If non-trivial generic goal is achieved with sufficiently rich class of helpful servers, then it is safely achieved with every server.

# Conclusions

- Is there a universal communication protocol?
  - No! (All functions vs. PSPACE-computable functions).
  - But can achieve sensible goals universally.
  - But ... diversity of goals may be the barrier to universality.
- Goals of communication.
  - Should be studied more.
  - Suggests good heuristics for protocol design:
    - Server = Helpful?
    - User = Sensing?

# Language Learning

- Meaning = end effect of communication.
  - [Dewey 1920s, Wittgenstein 1950s]
- What would make learning more efficient?
  - What assumptions about "language"?
  - How to do encapsulate it as "class" restrictions on users/servers.
  - What learning procedures are efficient?
- Time to get back to meaningful conversation!

# References

- Juba & S.
  - ECCC TR07-084: <http://eccc.uni-trier.de/report/2007/084/>
- Goldreich, Juba & S.
  - ECCC TR09-075: <http://eccc.uni-trier.de/report/2009/075/>



**Thank You!**