
Generación de lenguaje natural

PARTE 2: Generación de instrucciones en un entorno virtual

VIERNES

Luciana Benotti & Carlos Areces

luciana.benotti@gmail.com

Equipo PLN – Universidad Nacional de Córdoba

INRIA Nancy Grand Est

Qué hicimos ayer?

- ♦ Introducción a los entornos virtuales
 - ♦ Una rápida historia
 - ♦ Generación de instrucciones en entornos virtuales
 - ♦ GIVE World: Un mundo virtual simple
- ♦ Herramientas para determinación del contenido
 - ♦ Introducción a planning, planning vs búsqueda
 - ♦ Representación de un problema de planning
 - ♦ El algoritmo de graphplan
 - ♦ Búsqueda en el espacio de estados y heurísticas

Qué hicimos ayer?

- ◆ Introducción a los entornos virtuales
- ◆ Herramientas para determinación del contenido

Que vamos a hacer hoy?

- ♦ Introducción a los entornos virtuales
- ♦ Herramientas para determinación del contenido
- ♦ Determinación del contenido como planning
- ♦ Generación de referencias en mundos virtuales
 - ♦ El corpus SCARE
 - ♦ El corpus GIVE
- ♦ Evaluación de sistemas de GLN
 - ♦ El GIVE Challenge
- ♦ Conclusiones del curso

Relación con el material

- ♦ Introducción a los entornos virtuales
- ♦ Herramientas para determinación del contenido *
- ♦ Determinación del contenido como planning
- ♦ Generación de referencias en mundos virtuales *
 - ♦ El corpus SCARE
 - ♦ El corpus GIVE
- ♦ Evaluación de sistemas de GLN *
 - ♦ El GIVE Challenge
- ♦ Conclusiones del curso

Qué vamos a hacer hoy?

- ♦ Introducción a los entornos virtuales
- ♦ Herramientas para determinación del contenido
- ♦ **Determinación del contenido como planning**
- ♦ Generación de referencias en mundos virtuales
 - ♦ El corpus SCARE
 - ♦ El corpus GIVE
- ♦ Evaluación de sistemas de GLN
 - ♦ El GIVE Challenge
- ♦ Conclusiones del curso

Recuerden: el mundo GIVE-2



Recursos para Determinar Contenido

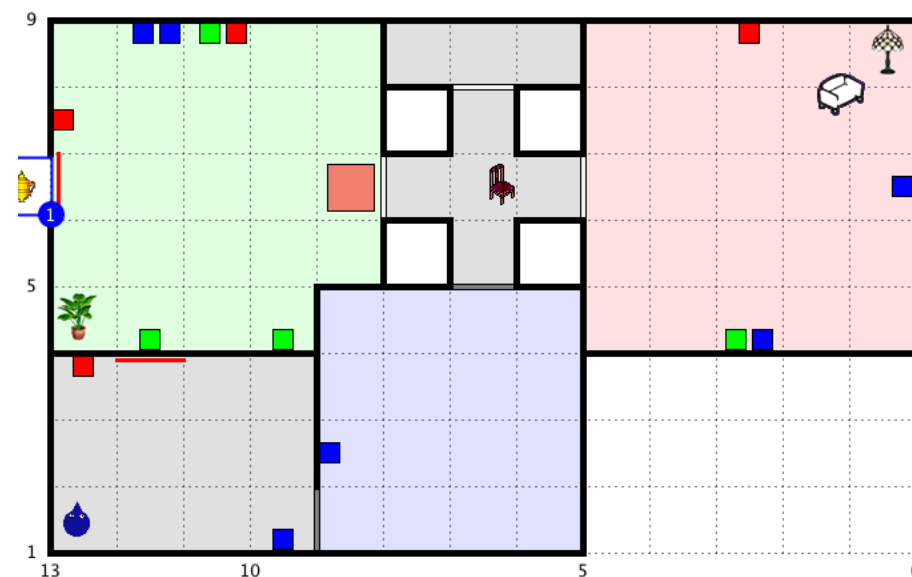
- ♦ Saber qué es lo que quiero lograr con las instrucciones:
 - ♦ Representar y razonar sobre **la meta**
- ♦ Saber cuál es la situación en la que el mundo se encuentra en este momento:
 - ♦ Representar y razonar sobre **el estado inicial**
- ♦ Saber cómo puede cambiar el mundo:
 - ♦ Representar y razonar sobre **las acciones** posibles
- ♦ Y un **planner!**

Recursos para Determinar Contenido

- La **meta**, eg `(not lost)and(holding trophy1)`
- El **estado**: el mundo GIVE-2
- Las **acciones**:
 - `move`
 - `push`
 - `take`
- **Planner**: `lazyff`

Recursos: El estado

- ♦ En GIVE-2 el estado es continuo
- ♦ Los planners sólo pueden manejar estados discretos
- ♦ Debo discretizar el estado.



Recursos: discretizando el estado



Las acciones: el ejemplo de *move*

:action move (?from - position ?to - position)

:precondition

(player-position ?from)

(adjacent ?from ?to non-blocked-state)

(alarm-state ?to non-alarmed-state)

:effect

(not (player-position ?from))

(player-position ?to)

Las acciones: el ejemplo de *push*

:action push (?b1 - object ?oldstate - other ?newstate - other ?pos - position ?y - object)

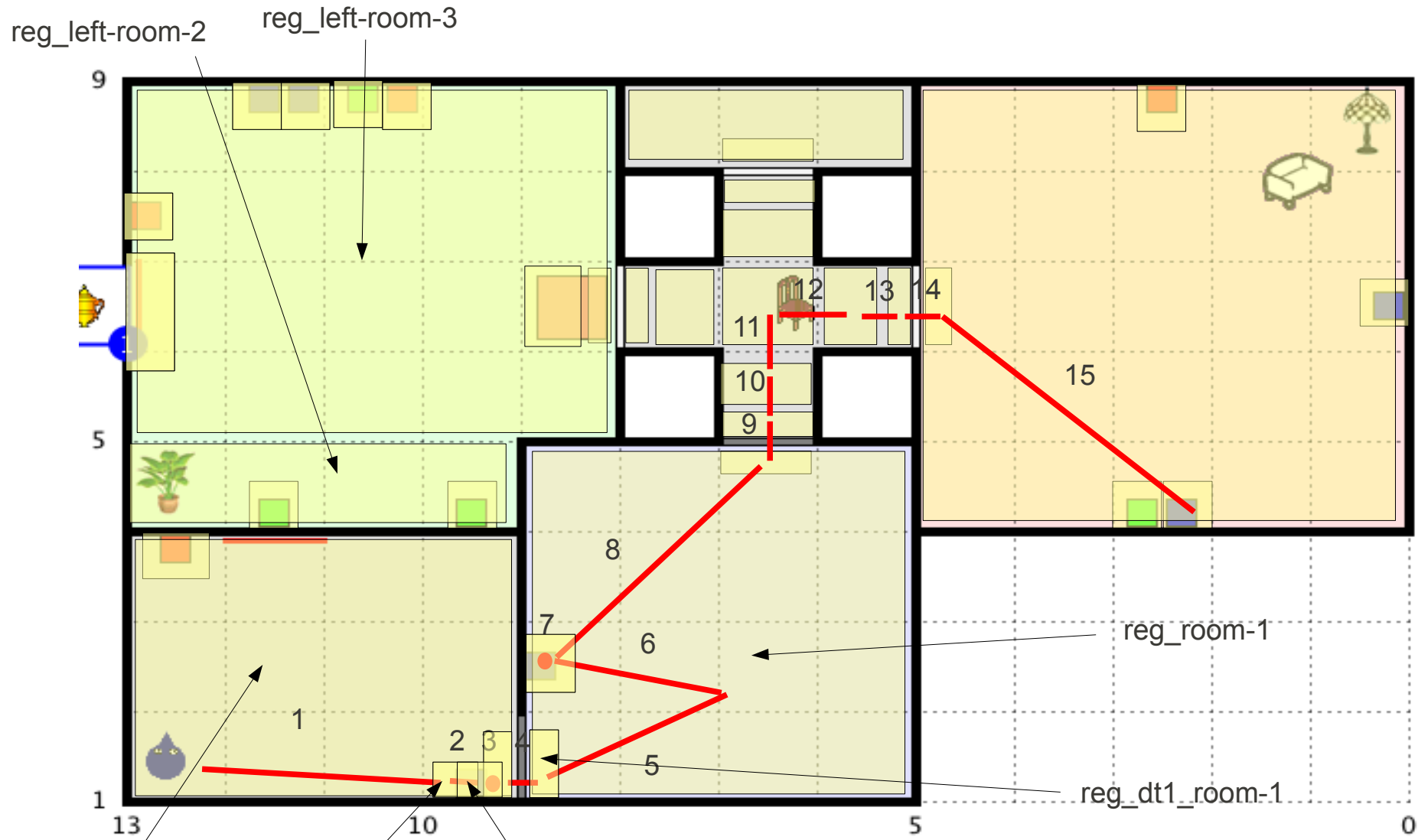
:precondition (and (activated ?b1) (activates ?b1 ?y) (button ?b1) (state ?b1 ?oldstate) (next-state ?oldstate ?newstate) (player-position ?pos) (near-object ?b1 ?pos))

:effect (and(activated ?y) (not (activated ?b1)) (not (state ?b1 ?oldstate)) (state ?b1 ?newstate) (forall (?d - object) (forall (?s - other) (forall (?ss - other) (when (and (controls ?b1 ?d) (state ?d ?s) (next-state ?s ?ss)) (and (not (state ?d ?s)) (state ?d ?ss)))))) (forall (?d - object) (forall (?bs1 - other) (forall (?bs2 - other) (forall (?bp1 - position) (forall (?bp2 - position) (when (and (controls ?b1 ?d) (connects ?d ?bp1 ?bp2) (adjacent ?bp1 ?bp2 ?bs1) (next-state ?bs1 ?bs2)) (and (not (adjacent ?bp1 ?bp2 ?bs1)) (adjacent ?bp1 ?bp2 ?bs2)))))))) (forall (?a - object) (forall (?as1 - other) (forall (?as2 - other) (forall (?ap1 - position) (when (and (controls ?b1 ?a) (alarm ?a) (not (door ?a)) (near-object ?a ?ap1) (alarm-state ?ap1 ?as1) (next-state ?as1 ?as2)) (and (not (alarm-state ?ap1 ?as1)) (alarm-state ?ap1 ?as2)))))) (forall (?x - object) (forall (?p1 - position) (forall (?p2 - position) (when (and (controls ?b1 ?x) (picture ?x) (position ?x ?p1)) (and (not (position ?x ?p1)) (position ?x ?p2)))))) (forall (?x - object) (forall (?p1 - orientation) (forall (?p2 - orientation) (when (and (controls ?b1 ?x) (picture ?x) (orientation ?x ?p1) (state-change-effect-turn ?b1 ?newstate ?x ?p2)) (and (not (orientation ?x ?p1)) (orientation ?x ?p2)))))))))

Un plan en GIVE-2

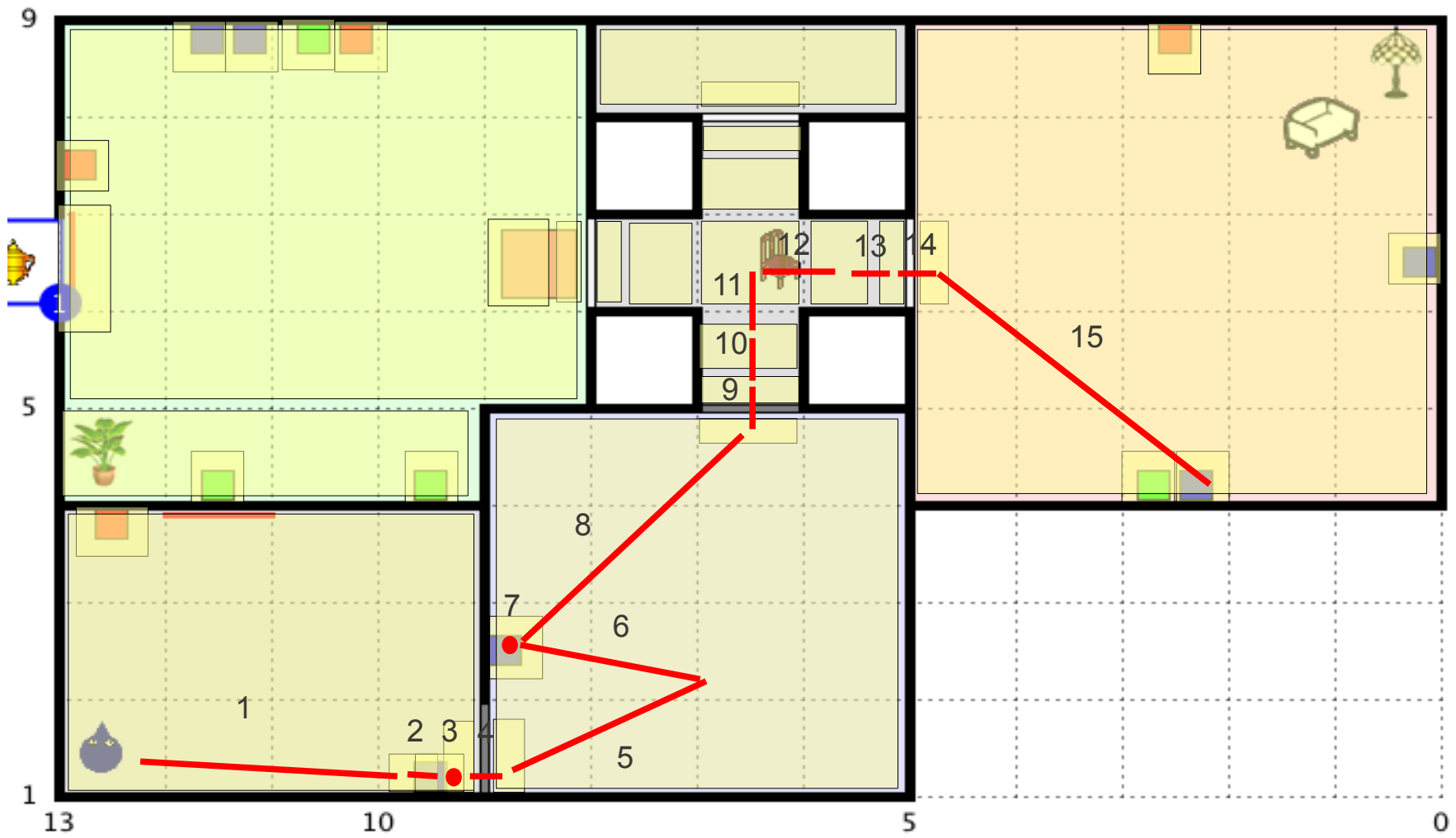
- ◆ 0.001: (MOVE REG_TUTORIAL-ROOM REG_DT1_TUTORIAL-ROOM) [1]
- ◆ 1.002: (MOVE REG_DT1_TUTORIAL-ROOM REG_DT1_ROOM-1) [1]
- ◆ 2.003: (MOVE REG_DT1_ROOM-1 REG_ROOM-1) [1]
- ◆ 3.004: (MOVE REG_ROOM-1 REG_B1_ROOM-1) [1]
- ◆ 4.005: (PUSH B1 OFF ON REG_B1_ROOM-1 B5) [1]
- ◆ 5.006: (MOVE REG_B1_ROOM-1 REG_ROOM-1) [1]
- ◆ 6.007: (MOVE REG_ROOM-1 REG_D2_ROOM-1) [1]
- ◆ 7.008: (MOVE REG_D2_ROOM-1 REG_CORRIDOR1B-3_D2) [1]
- ◆ 8.009: (MOVE REG_CORRIDOR1B-3_D2 REG_CORRIDOR1B-3) [1]
- ◆ 9.010: (MOVE REG_CORRIDOR1B-3 REG_CORRIDOR1B) [1]
- ◆ 10.011: (MOVE REG_CORRIDOR1B REG_CORRIDOR1B-2) [1]
- ◆ ...
- ◆ ~60 acciones en un mundo simple, 300 acciones en un mundo complejo

Un plan en GIVE-2



64 actos básicos hacia la meta

1) Sistema baseline



64 instrucciones hacia la meta

1) Sistema baseline: Demo

- ♦ El monitoreo de la ejecución es muy restrictivo, el sistema no deja que el usuario se salga del plan
- ♦ En la demo veremos frecuentemente:
 - ♦ *Wait, that's not what I wanted you todo.*

NLG: Look around until you see a blue thing. Then go towards it.

User: [goes really close to the button]

NLG: Excellent! Press the button.

User: [walks backwards a bit to center the button]

NLG: Wait. That's not what I wanted you to do. [replan] Go to the button.

User: [presses the button]

NLG: Wait. That's not what I wanted you to do. [replan] Go to the door.

Dando instrucciones mas flexibles

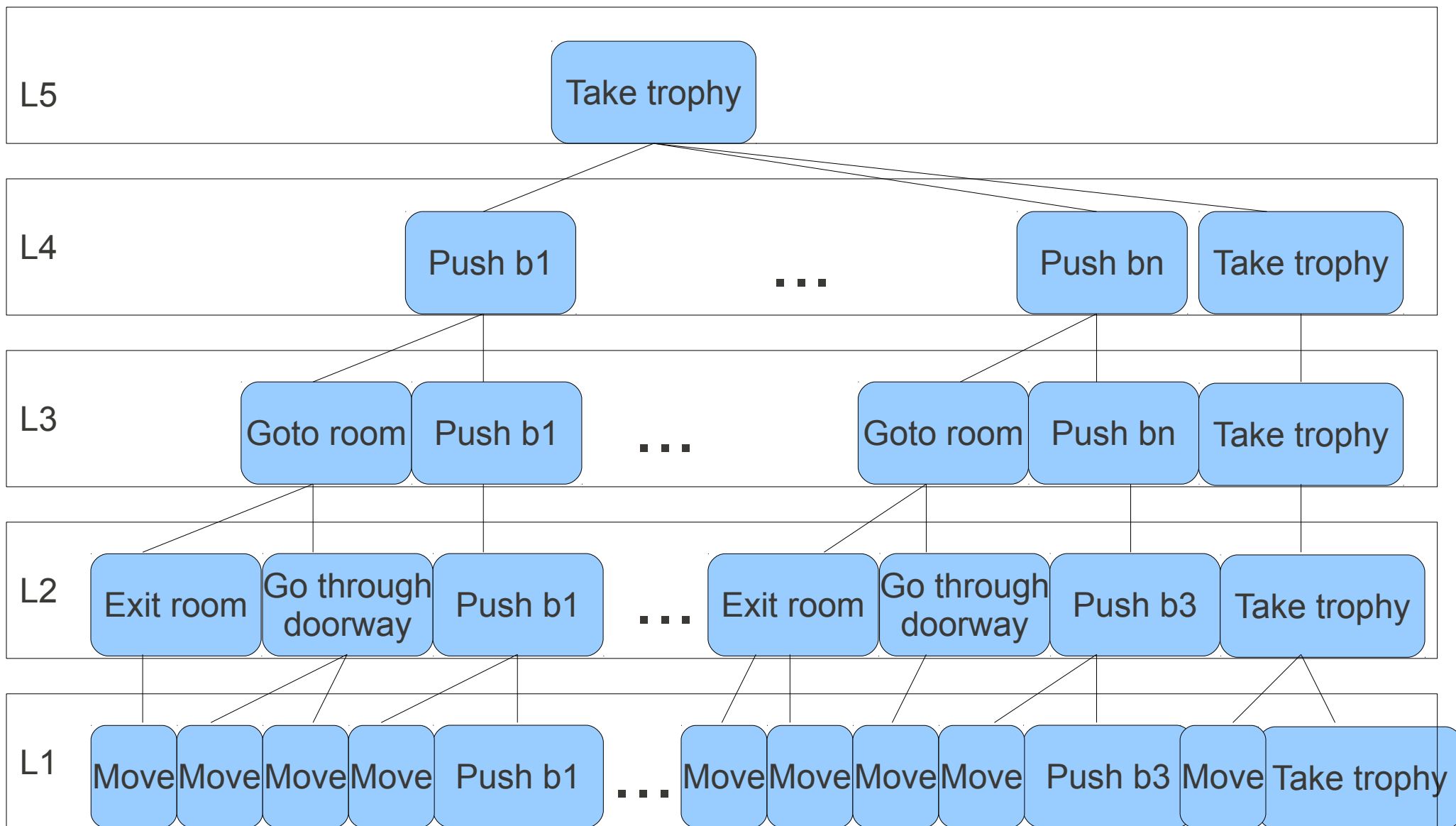
- ♦ Puedo dar instrucciones más flexibles dando instrucciones de granularidad más gruesas.
- ♦ Dos instrucciones tiene diferente granularidad si:
 - ♦ Las dos intentan lograr la misma meta
 - ♦ Pero una hace explícitos actos comunicativos que son tácitos (implícitos) en la otra
- ♦ Una instrucción con granularidad más gruesa da más libertad al DF porque él es libre de decidir cómo ejecutar las acciones tácitas

5 granularidades en GIVE

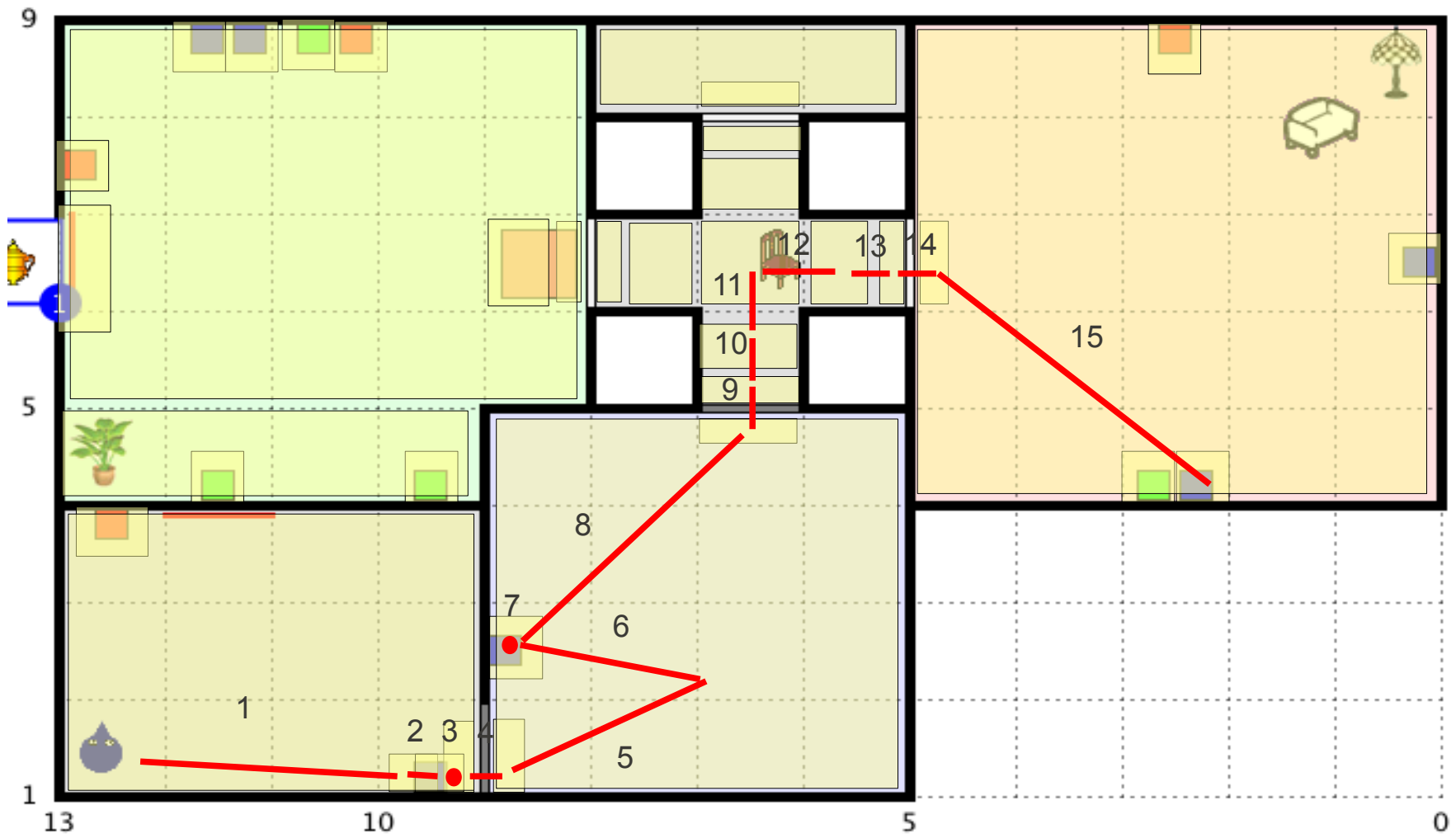
Ya vimos una posible granularidad (verbalizar cada acción). Ahora veremos 4 granularidades mas gruesas

- 1) **Ninguna acción es tácita** (sistema baseline)
- 2) **Algunas acciones move son tácitas** (sistema NA)
- 3) **La mayoría de las acciones move son tácitas** (sistema NM)
- 4) **Todas las acciones move** son tácitas
- 5) **Todas las acciones move y push** son tácitas

5 granularidades en GIVE

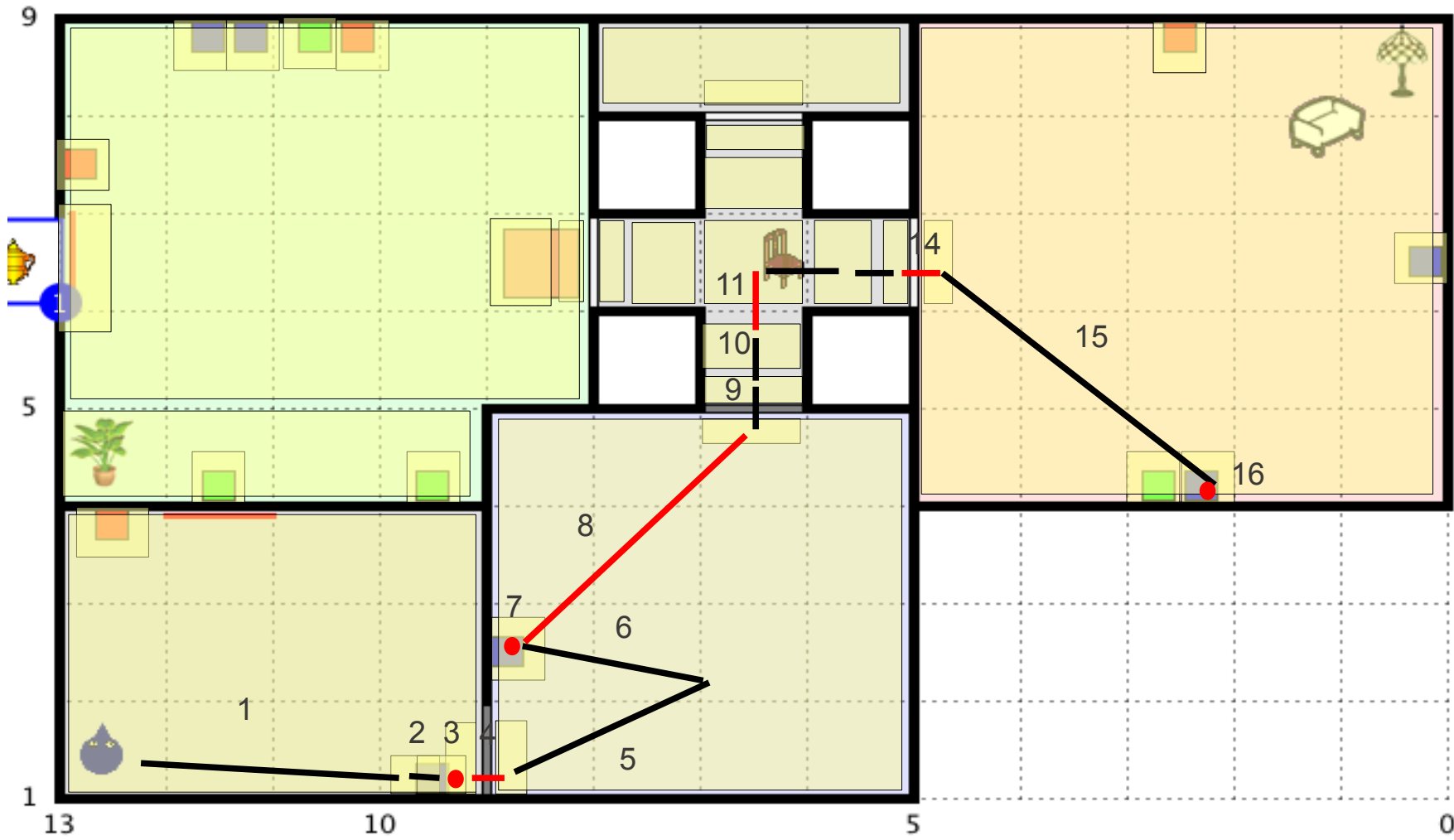


1) Sistema baseline



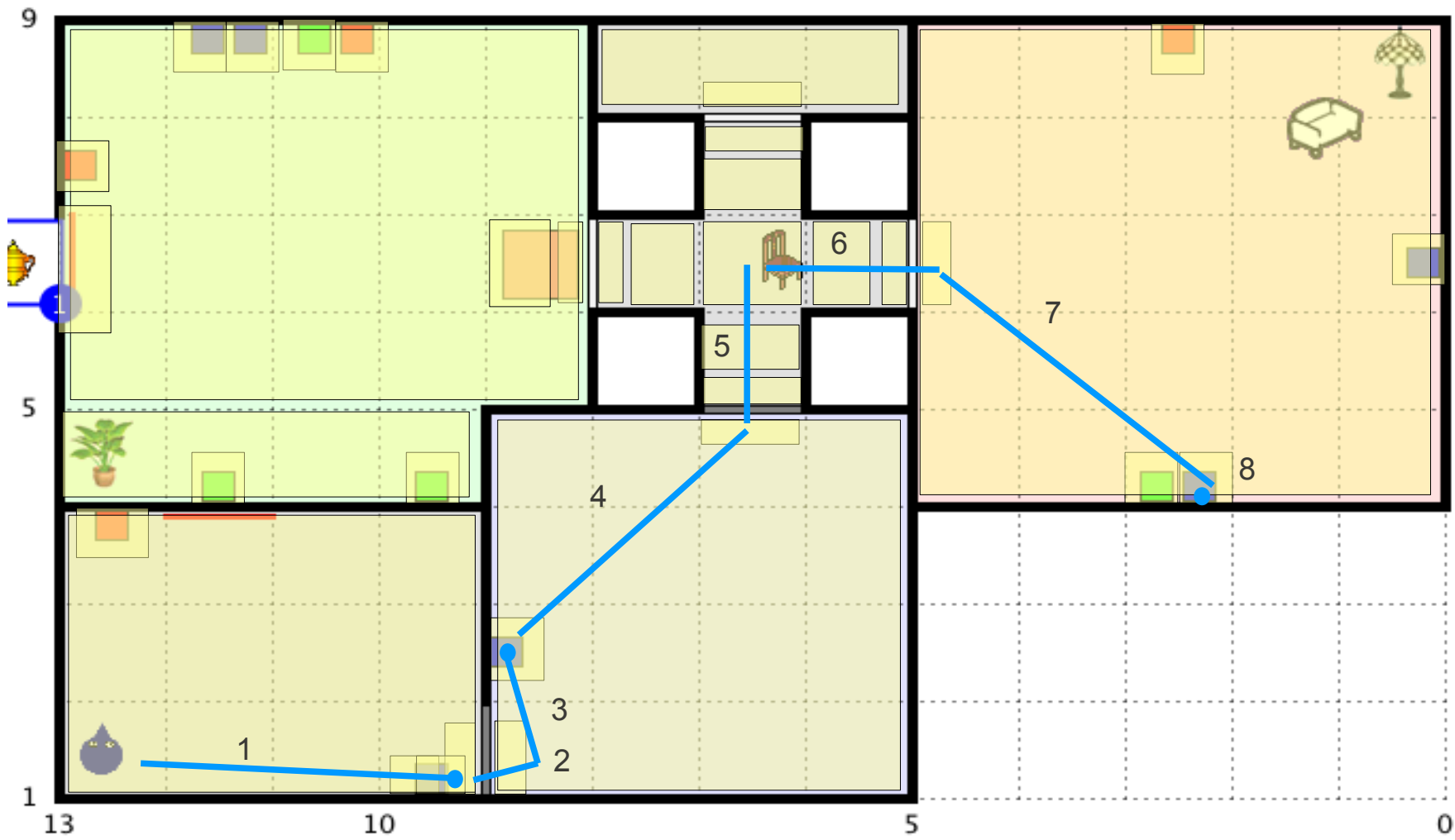
64 instrucciones a la meta

2) Sistema NA (Denis, et al 2010)



Algunas instrucciones move son tácitas

2) Sistema NA (Denis, et al 2010)



~25 instrucciones hasta la meta

2) Sistema NA: Demo

NLG: Push the blue button. Yeah! This one!

User: [goes really close to the button]

User: [walks backwards a bit to center the button]

User: [presses the button]

NLG: Perfect! Exit the room by going forward.

User: [goes through the door]

NLG: Push the button. Yeah! This one!

User: [presses the button]

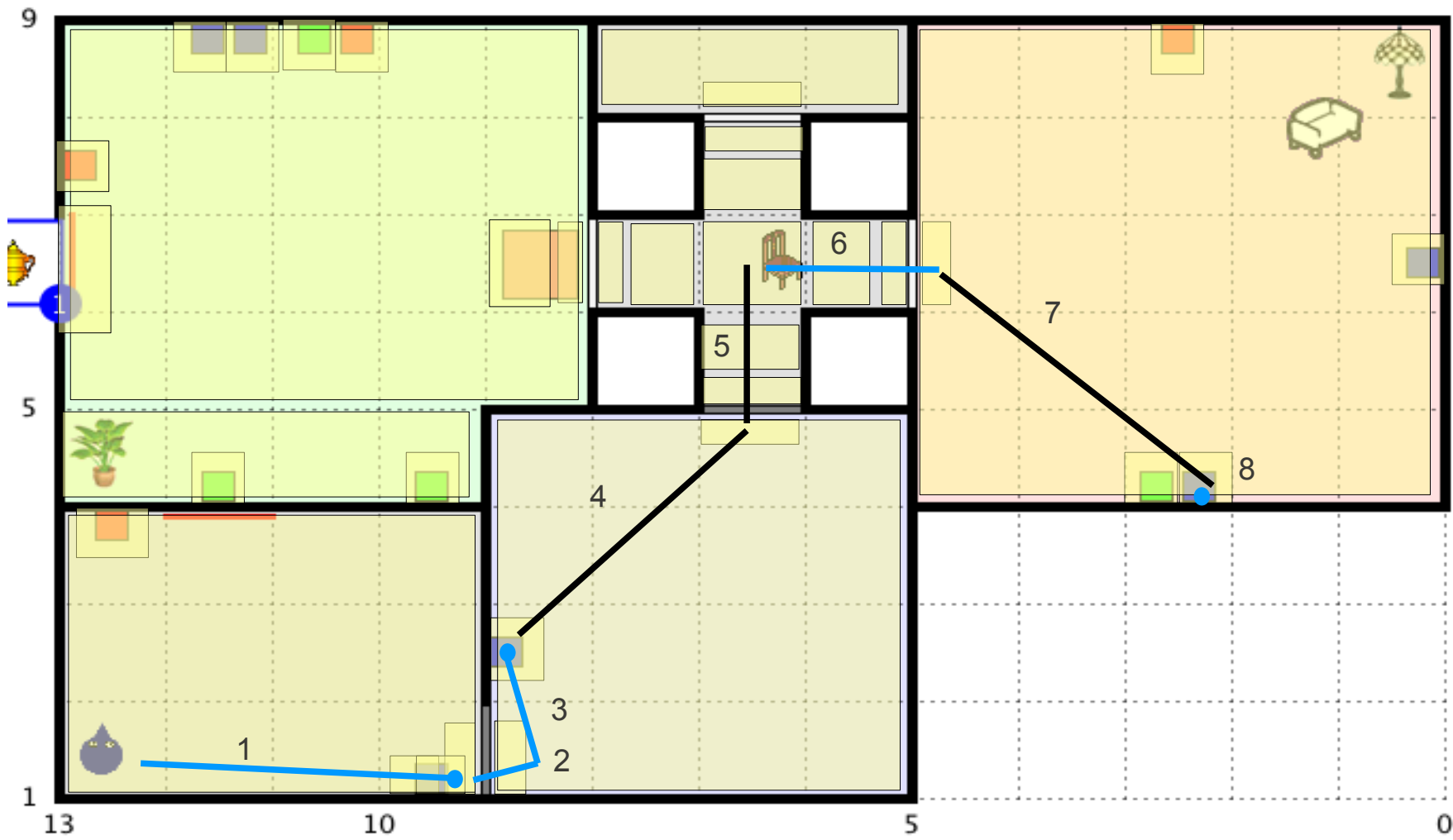
NLG: Exit the room through the doorway behind you.

User: [goes back to the tutorial room]

*NLG: No, no, Wait. **[replan]** Go through the doorway in front of you.*

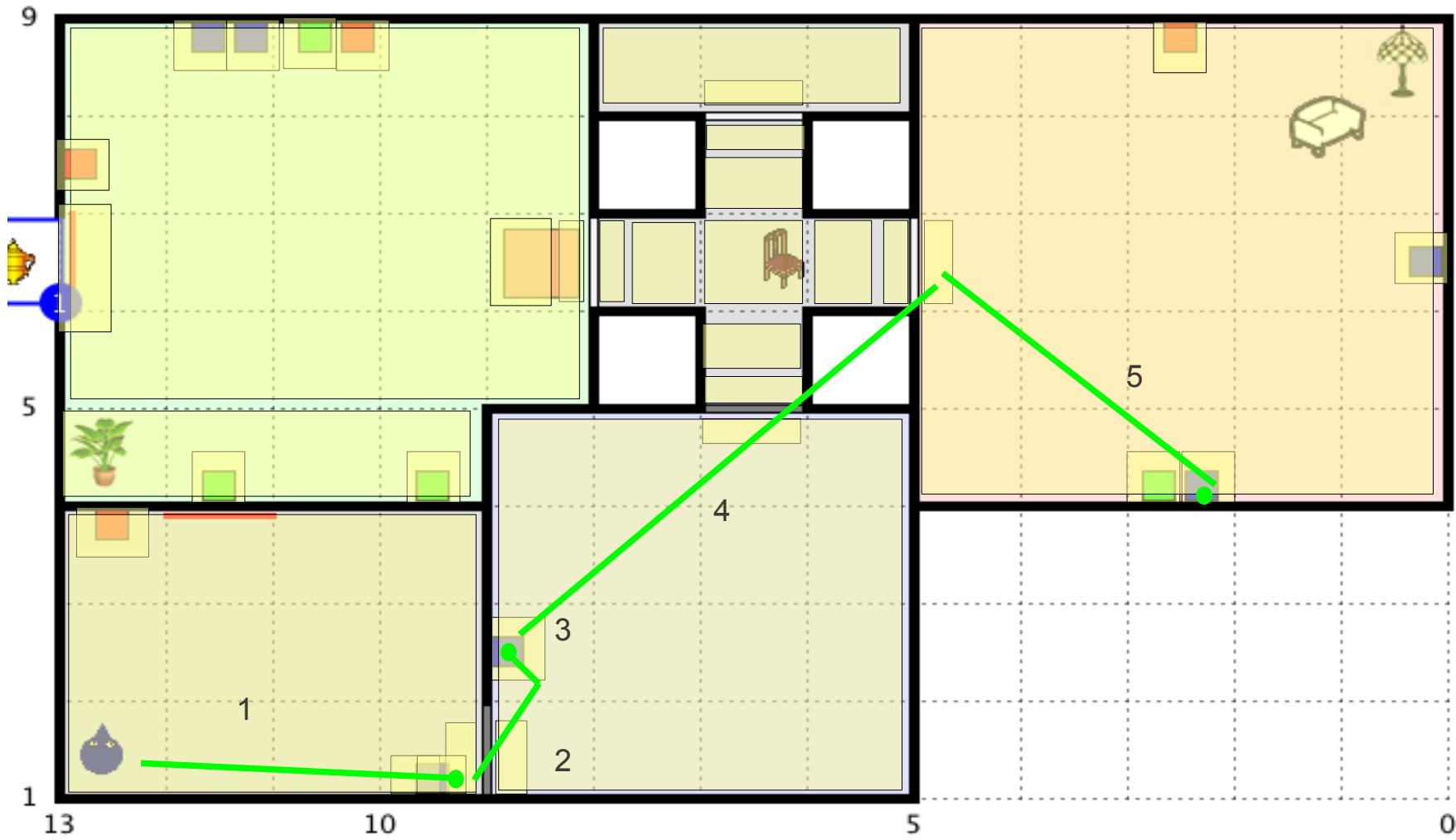
- No hay replanning cuando el usuario se sale del plan
- El sistema replanea si el jugador para por una puerta incorrecta, es decir, cuando el jugador se sale del plan en nivel 2

3) Sistema NM (Amoia et al, 2010)



La mayoría de los move son tácitos

3) Sistema NM (Amoia et al, 2010)



~15 instrucciones a la meta

3) Sistema NM: Demo

NLG: To open the door push the blue button on your right.

User: [presses the button]

NLG: Great! This was the right button. Look for the blue room.

User: [goes through the door]

NLG: Great! To open the door press the blue button on your left.

User: [presses the button]

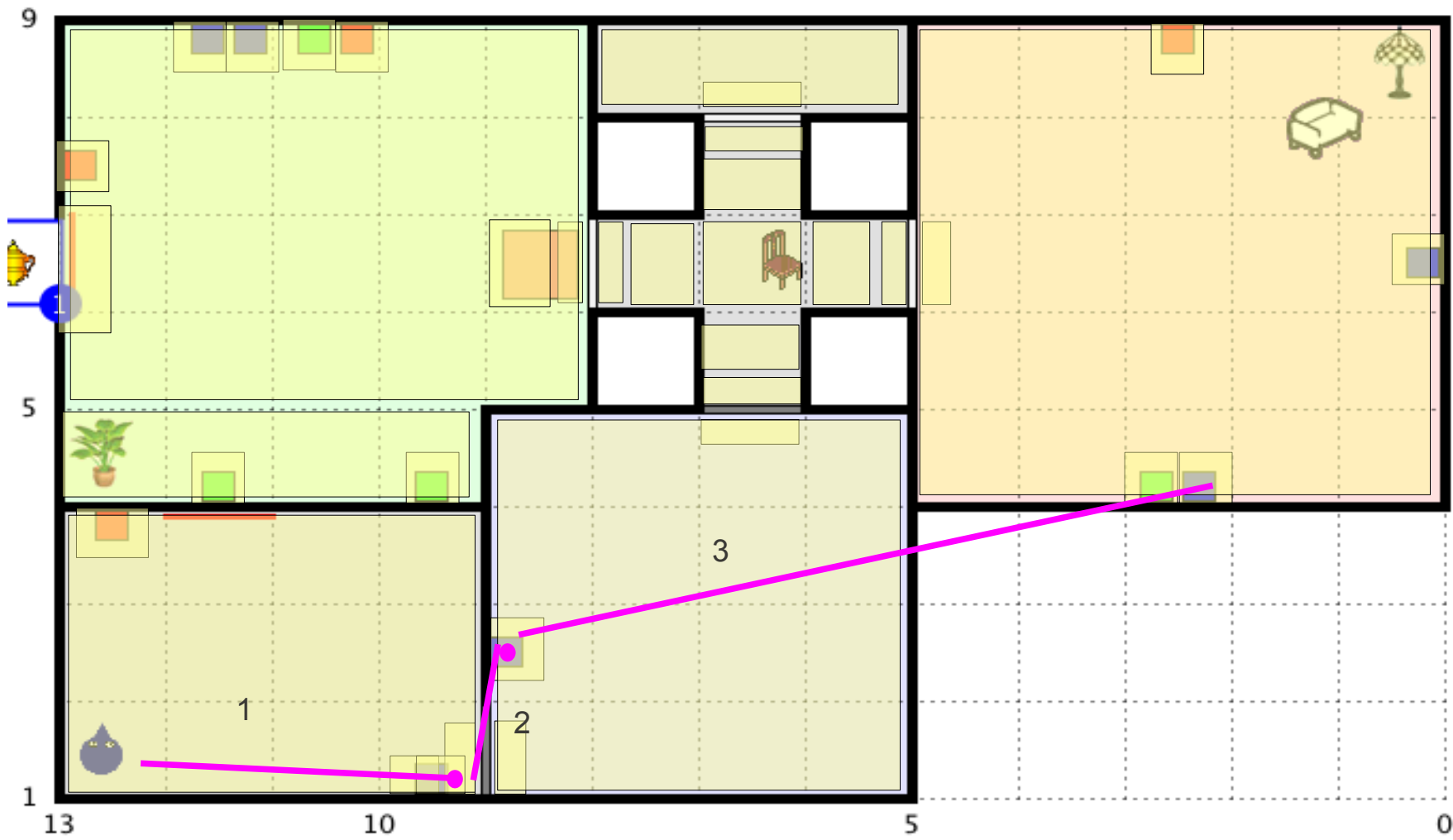
NLG: Great! This was the right button. Look for the room with the lamp.

User: [goes back to the tutorial room]

NLG: [says nothing until the user gets to the room with the lamp]

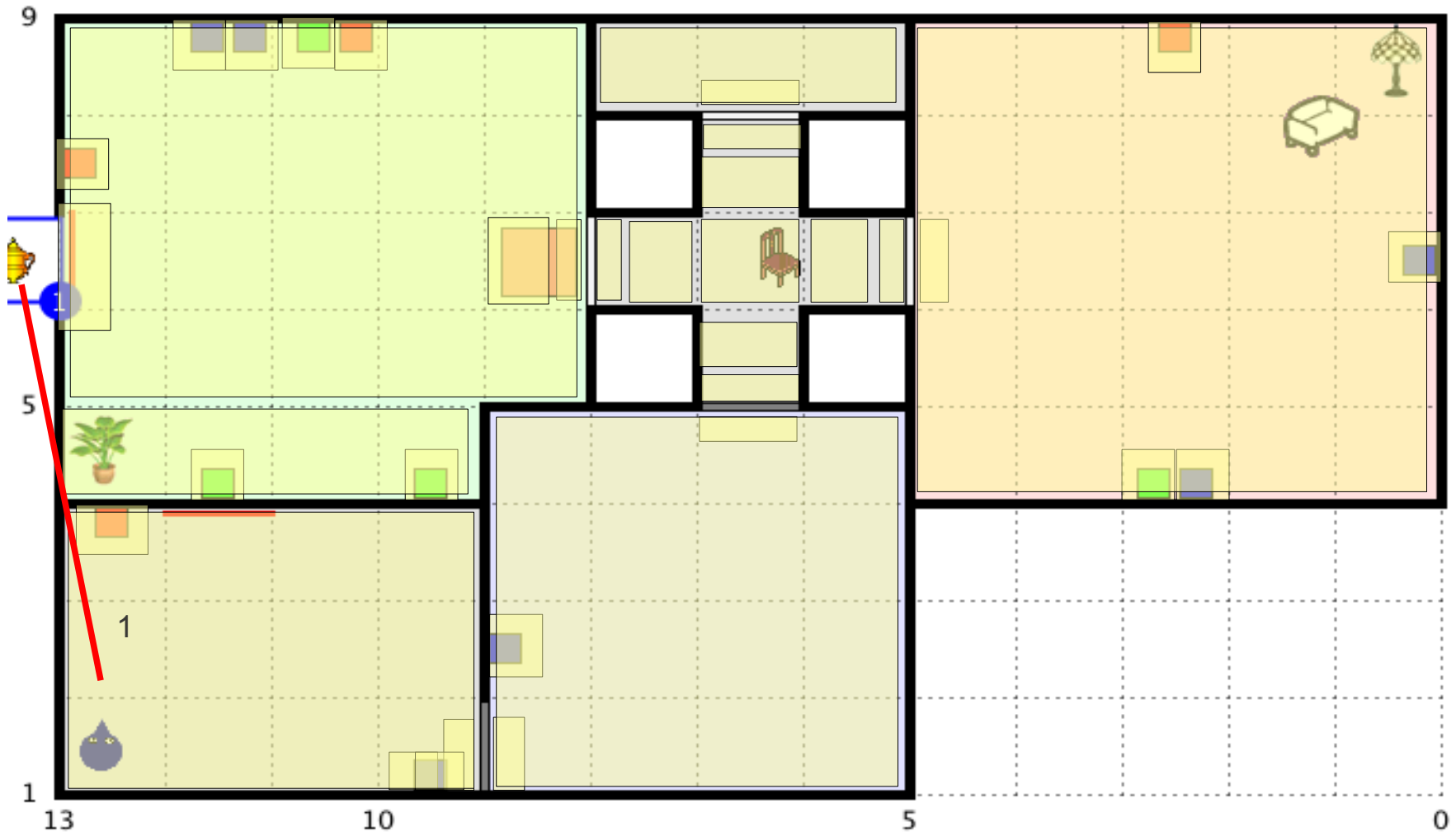
- El sistema no replanifica aún si el jugador ingresa a la habitación incorrecta
- El sistema replanifica si el usuario aprieta el botón incorrecto, es decir, el sistema se sale del plan en el nivel 3

4) Granularidad sólo push and take



10 instrucciones a la meta

5) Granularidad “Sólo take”



1 paso a la meta !

La jerarquía de granularidad

All moves and pushes tacit
Level L5

Take trophy

All moves tacit
L4

Push b1

...

Push bn

Take trophy

Marilisa
L3

Goto room

Push b1

...

Goto room

Push bn

Take trophy

Alex
L2

Exit room

Go through doorway

Push b1

...

Exit room

Go through doorway

Push b3

Take trophy

Plan
L1

Move

Move

Move

Move

Push b1

...

Move

Move

Move

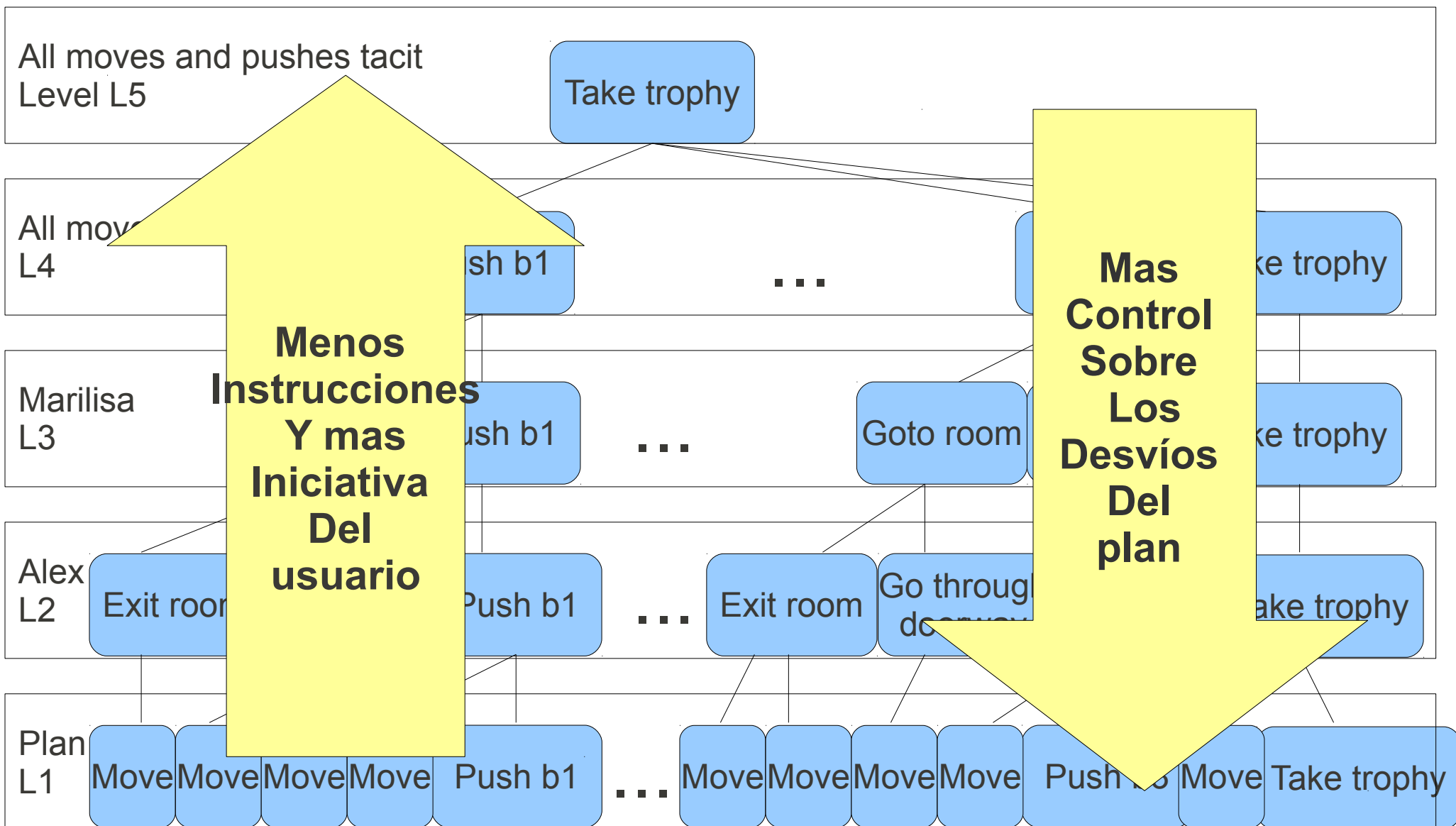
Move

Push b3

Move

Take trophy

Efectos de cambiar de granularidad



Sistemas inter-granularidad

- ♦ Los tres sistemas que mostramos dan instrucciones a la misma granularidad a la que la monitorean
- ♦ Pero un sistema podría generar instrucciones en el nivel L_i y monitorear en el nivel L_{i-1}
- ♦ Si el sistema detecta problemas en L_{i-1} puede ofrecer clarificaciones o cambiar a una política de un nivel más bajo

Sistemas inter-granularidad

- ◆ L4 -> L3

- ◆ *NLG: We have to press the button that dis-activates the alarm*
- ◆ *User: [goes to the green room]*
- ◆ *NLG: It's in the red room*

- ◆ L3 -> L2

- ◆ *NLG: Return to the room with the lamp*
- ◆ *User: [goes through the wrong door]*
- ◆ *NLG: It's not this way. Go back.*

Cúando bajar en la jerarquía?

Idea hasta ahora:

- ♦ Cuando el jugador hace una acción incorrecta.

Otras ideas:

- ♦ Cuando el DF lo solicita
- ♦ En puntos de decisión (intersecciones)
- ♦ Cuando está llevando “demasiado tiempo” hacer la acción indicada (eg. 3 segundos?)
- ♦ Cuándo hay “titubeo”
- ♦ Cuando la instrucción es “muy pesada cognitivamente”
 - ♦ “hit the left red button in the right wall in the next room in front of you”

Resumiendo ...

- ♦ Es necesario hacer abstracciones sobre un plan antes de transmitirlo
- ♦ No hace falta incluir en el contenido a generar los mensajes que puedan ser inferidos (sin exploración); eg. *Moverse a un botón visible*
- ♦ Puedo decidir no incluir mensajes cuya inferencia requiere exploración; eg. *Buscar una habitación con una lámpara*
- ♦ Pero corro el riesgo de que el DF pierda el tiempo haciendo acciones innecesarias

Qué vamos a hacer hoy?

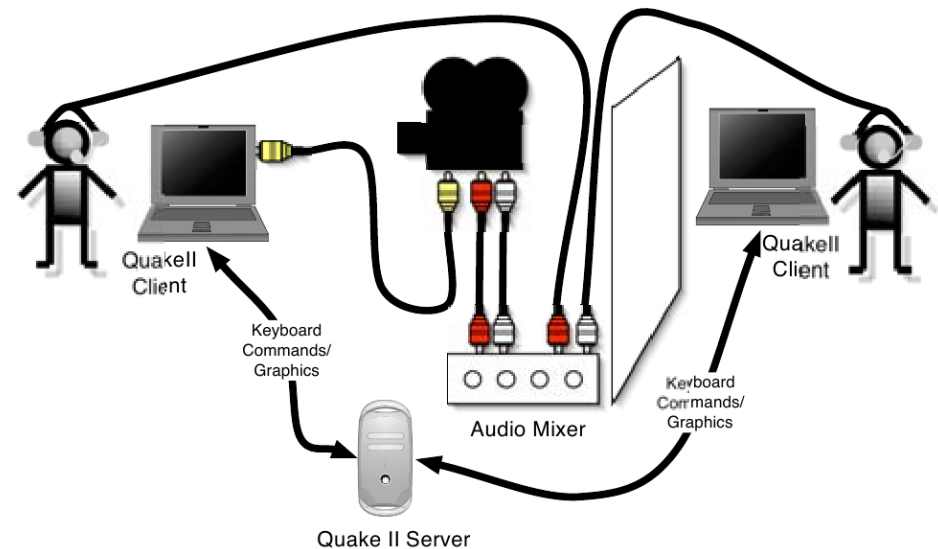
- ♦ Introducción a los entornos virtuales
- ♦ Herramientas para determinación del contenido
- ♦ Determinación del contenido como planning
- ♦ **Generación de referencias en mundos virtuales**
 - ♦ El corpus SCARE
 - ♦ El corpus GIVE
- ♦ Evaluación de sistemas de GLN
 - ♦ El GIVE Challenge
- ♦ Conclusiones del curso

Porqué GER situada es diferente?

- ♦ El entorno (el modelo) está cambiando; el DG puede dar instrucciones que cambien el modelo con el objetivo de generar una ER más simple
- ♦ La GER no es una frase nominal sino un proceso que puede tener varios turnos
- ♦ Las ER se puede generar incrementalmente, restringiendo el contexto poco a poco

GER en entornos virtuales

- ◆ Hace muy poco que se investiga GER en entornos virtuales, y GER situada en general
- ◆ El primer trabajo en esta dirección fue el de (Stoia, et al, 2007).
- ◆ Su primer paso fue recolectar un corpus de GER en entornos virtuales.
- ◆ El corpus SCARE



El corpus SCARE



Vista del mundo virtual del DF, también mostrada en el monitor del DG

- ♦ 15 diálogos humano-humano
- ♦ Situados en un mundo virtual de QUAKE
- ♦ El direction giver (DG) da instrucciones al direction follower (DF)
- ♦ El objetivo es llevar a cabo una tarea en el mundo virtual

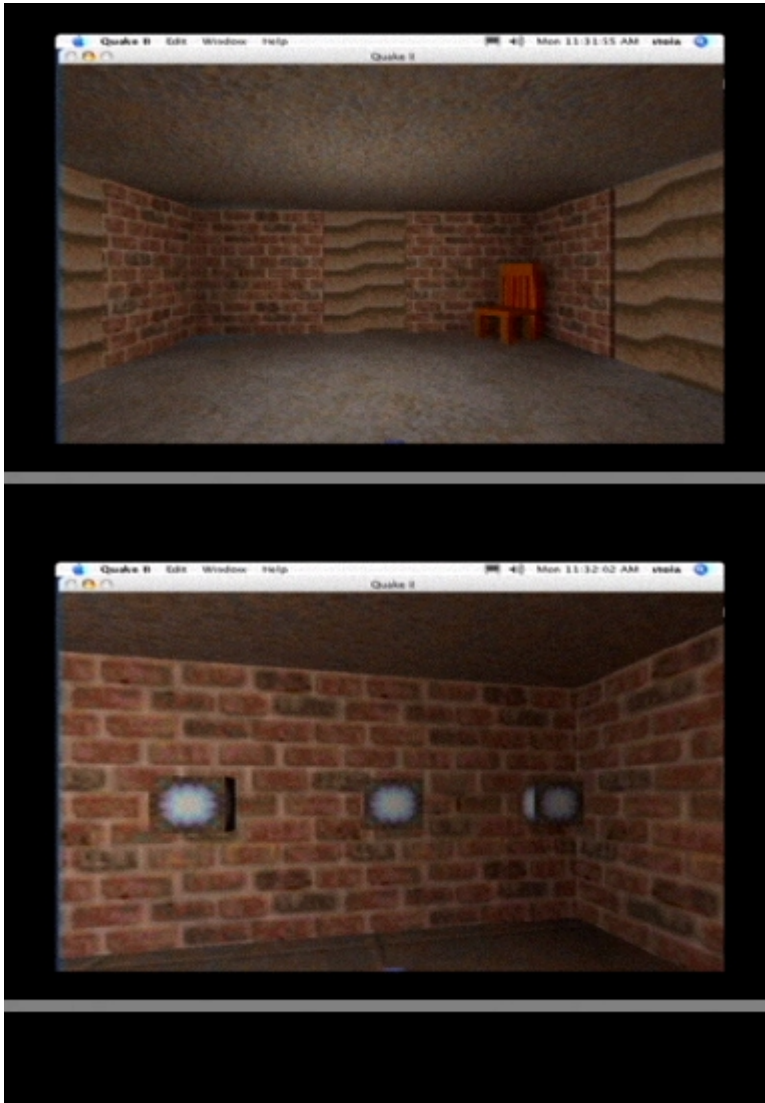
Ejemplo del corpus SCARE

Ejemplo paso a paso



- ◆ DG: alright
- ◆ DF: alright
- ◆ DG: ok, ok, [mumble], ok
- ◆ DG: go back to where you were facing
- ◆ DG: ok, so go through *the door*
- ◆ DF: ok

Ejemplo paso a paso



- ◆ DG: *so we have to move a picture to the other side of the wall or something*
- ◆ DG: *and then go through **the next one***
- ◆ DG: *ok, **there's three buttons, hit the middle one***

Ejemplo paso a paso



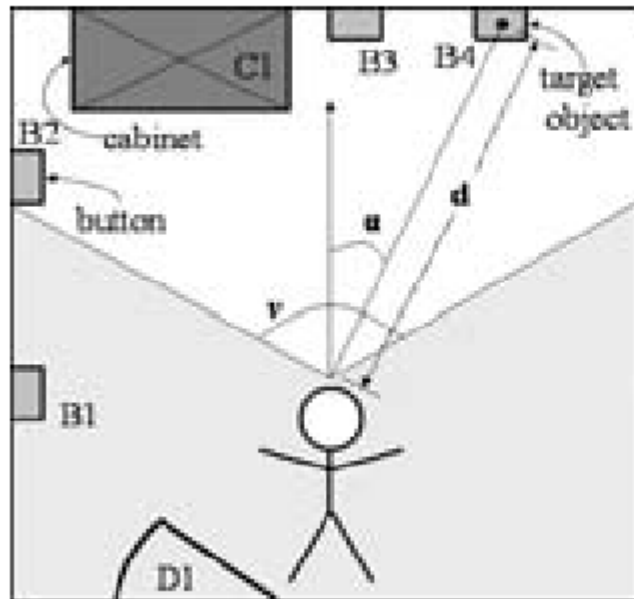
- ◆ DG: and that should have moved the picture
- ◆ DF: yeah, moved from the wall the buttons were on to the wall on the left

Enseñanzas del corpus SCARE

- ♦ La posición del DF con respecto al target es crucial:
 - ♦ Aparecen ERs deícticas
 - ♦ DG: Press **that**
 - ♦ El DG puede cambiar el contexto a propósito para generar una RE más simple.
 - ♦ DG: Turn left, press **that button**
- ♦ La RE no necesita ser “univoca”, puede ser en “cuotas”:
 - ♦ DG: We have to press a blue button in this room
 - ♦ DF: [Turns until he sees a blue button]
 - ♦ DG: Yeah, **that one**

Resultados sobre el corpus SCARE

- ♦ El experimento SCARE fue diseñado para investigar **deixis** (eg. *Press that*)
- ♦ Se eligieron *features* del contexto y se entrenó un árbol de decisión sobre cuándo usar deixis



$v = \text{Visible area}(100^\circ)$

$\alpha = \text{Angle to target}$

$d = \text{distance to target}$

In this scene:

VisDistr = 3 {B2, B3, C1}

VisSemDistr = 2 {B2, B3}

Resultados sobre el corpus SCARE

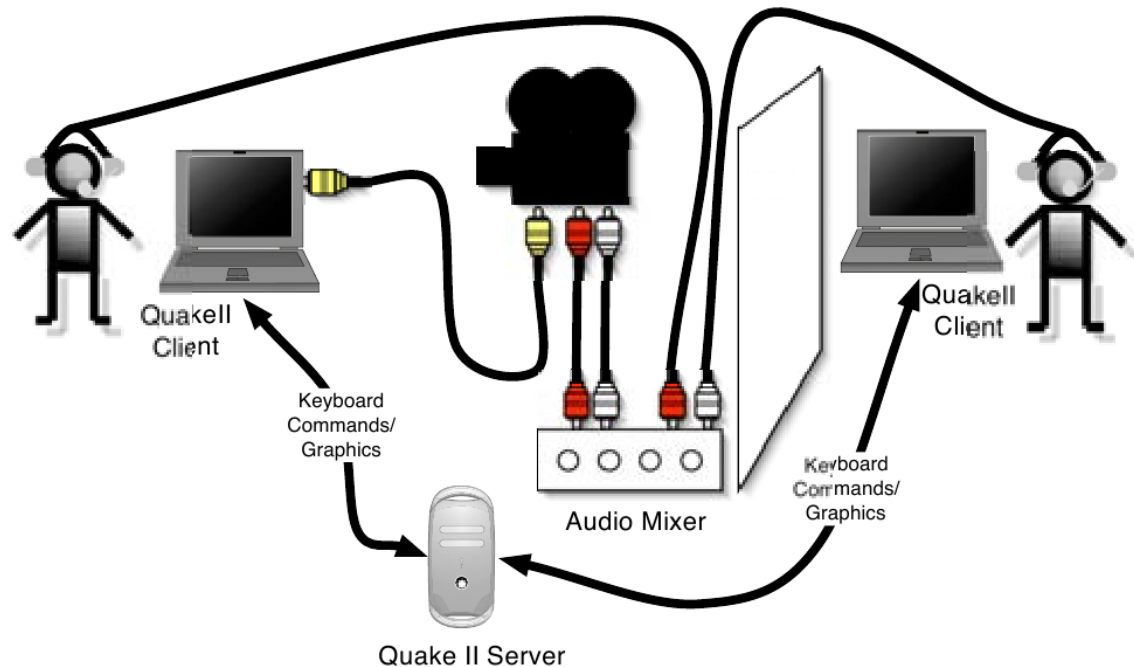
- ♦ El 62,6% de las expresiones generadas por el sistema fueron mejores o iguales que las humanas

System compared to Human	Trials: 577
equal	45.9%
system preferred	16.6%
(system equal or preferred to human)	(62.6%)
human preferred	37.4%

- ♦ Estos resultados fueron evaluados sobre 4 corpus de evaluación (11 para entrenamiento)
- ♦ Pero sobre el mismo mundo!

Mas alla de un mundo ...

- ◆ Y porqué no lo hacemos en otro mundo?
- ◆ Porque es mucho trabajo!
- ◆ Recolección ad-hoc
- ◆ Diseñar mundo
- ◆ Transcribir
- ◆ Anotar



El wizard GIVE

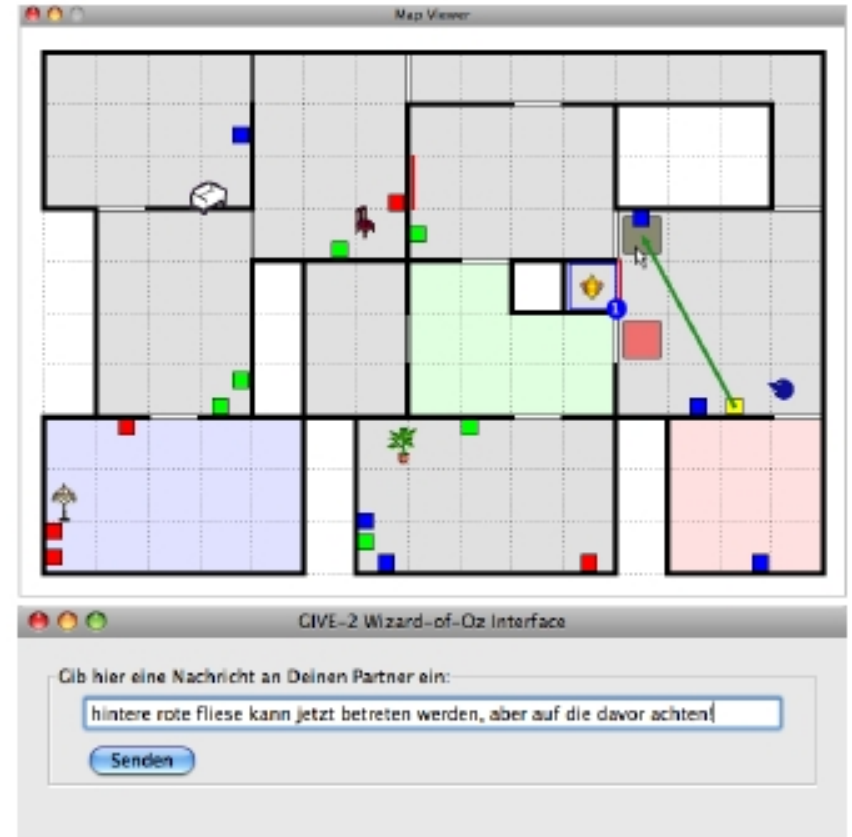


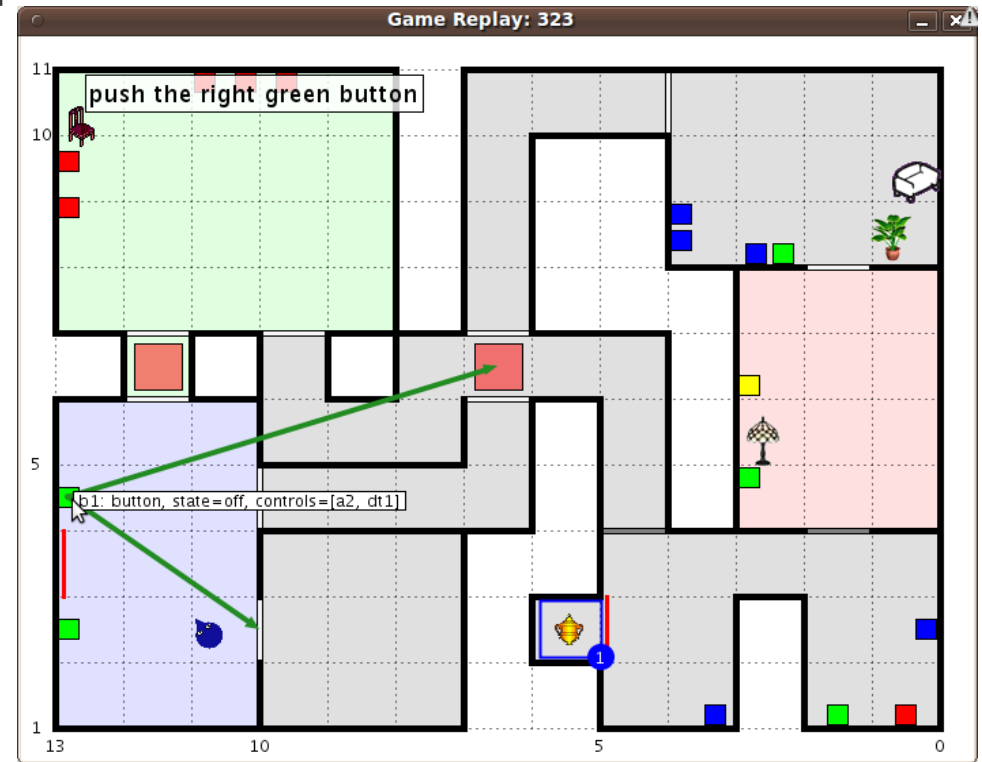
Figure 1: The view of the virtual environment, as displayed on the IF's monitor.

El wizard de GIVE

- ◆ El wizard de GIVE ofrece:
 - ◆ Un entorno de recolección de corpus
 - ◆ El diseño de nuevos mundos es simple, hay un generador de mundos random
 - ◆ La interacción es basada en texto, no en speech
 - ◆ La anotación es parcialmente automatizada (e.g., objetos visibles, interacciones con objetos, ingreso a una nueva habitación, etc)

El Corpus GIVE

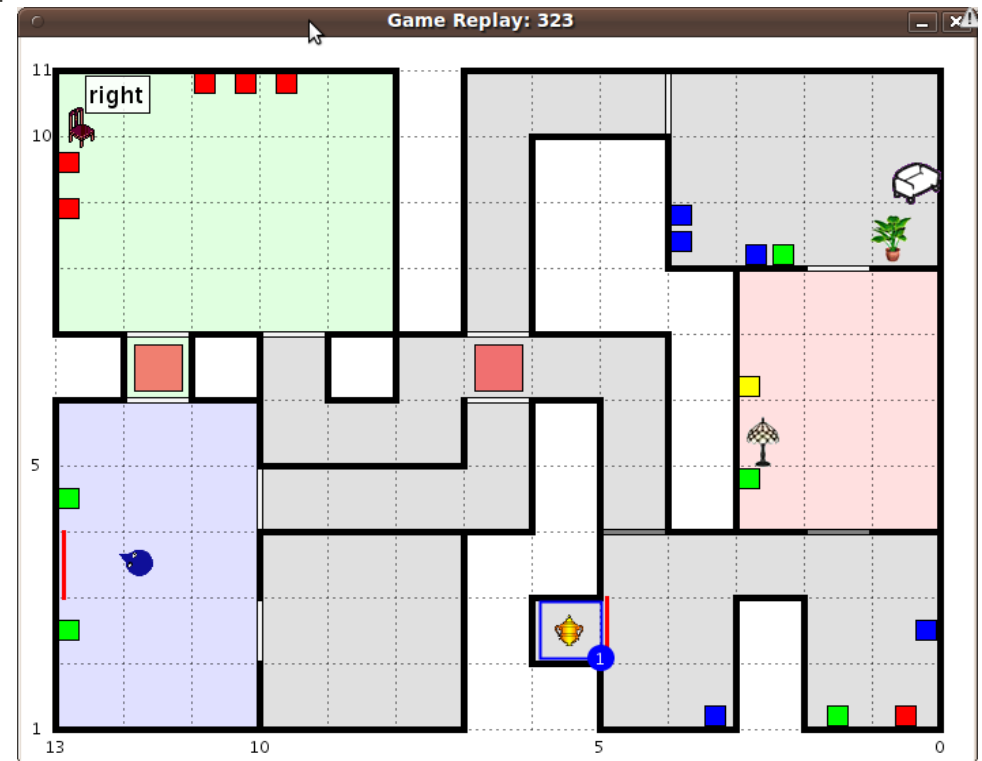
- ♦ El primer corpus GIVE fue recolectado por (Koller, et al 2010)
- ♦ Hay corpus en inglés y en alemán
- ♦ Anotación automática de datos (e.g. intercalado de acciones físicas y lingüísticas)



Visualizador del corpus

El Corpus GIVE

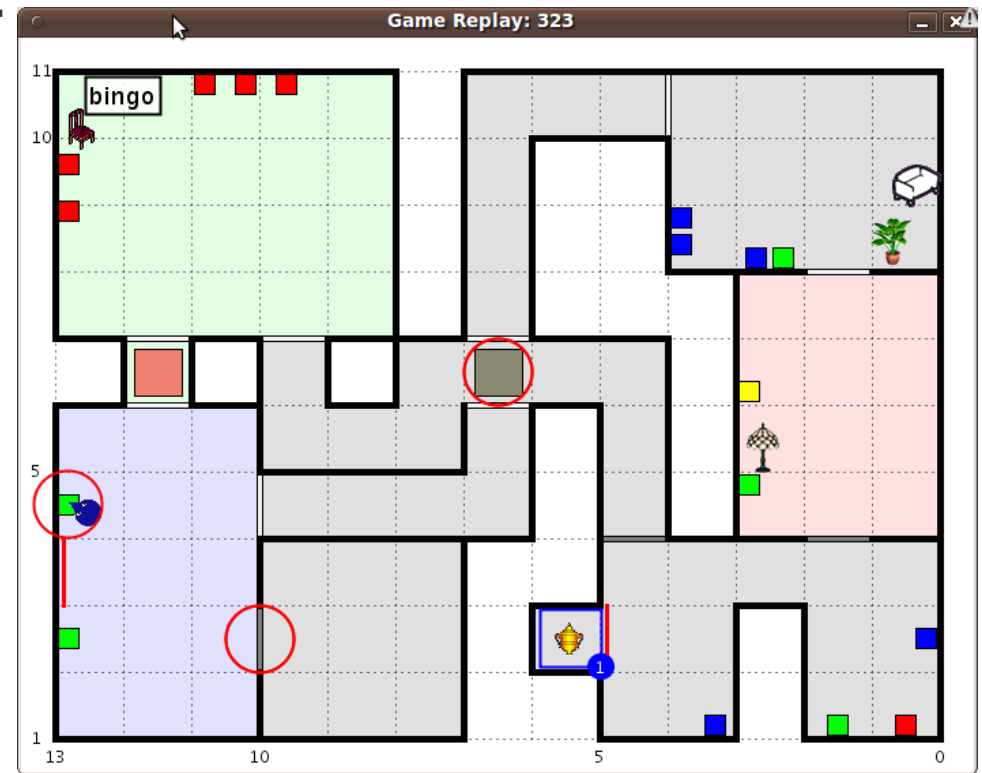
- ♦ El primer corpus GIVE fue recolectado por (Koller, et al 2010)
- ♦ Hay corpus en inglés y en alemán
- ♦ Anotación automática de datos (e.g. intercalado de acciones físicas y lingüísticas)



Visualizador del corpus

El Corpus GIVE

- ♦ El primer corpus GIVE fue recolectado por (Koller, et al 2010)
- ♦ Hay corpus en inglés y en alemán
- ♦ Anotación automática de datos (e.g. intercalado de acciones físicas y lingüísticas)



Visualizador del corpus

Enseñanzas del corpus GIVE

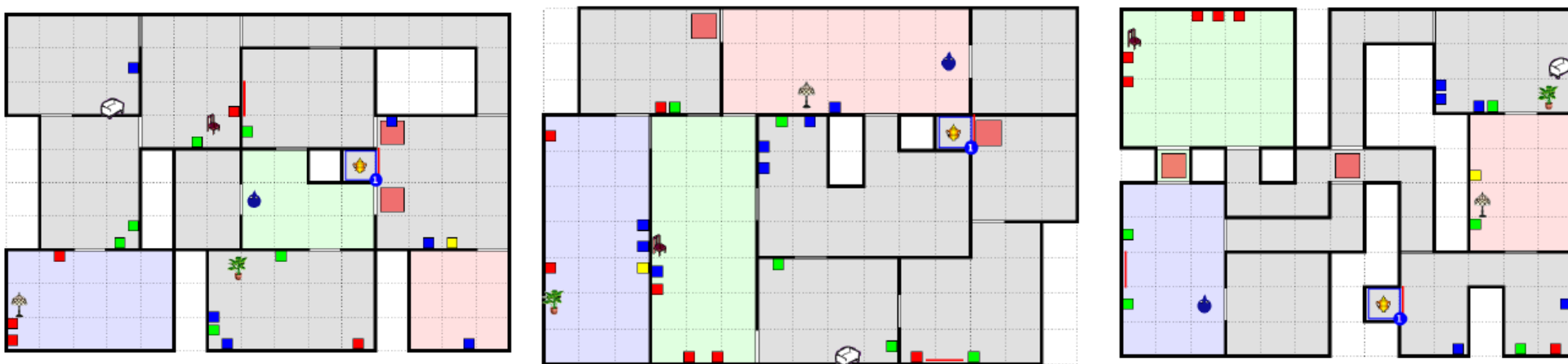
- ♦ Por el momento se han anotado los tipos de ER encontrados:
 - ♦ Propiedades taxonómicas: eg. “button”, “room”.
 - ♦ Propiedad absoluta: eg. “red”, “yellow”.
 - ♦ Propiedades relativas: e.g. “first”, “middle”.
 - ♦ Centrado en el usuario: e.g. “on the left”, “behind you”.

Enseñanzas del corpus GIVE

- ♦ Relación con landmark móvil (e.g. “by the chair”)
- ♦ Relacionado con un distractor (e.g. “next to the yellow button”)
- ♦ Relación con landmark inmóvil (e.g. “on the wall”)
- ♦ Historia de la interacción (e.g. “from before”, “as last time”)
- ♦ Foco visual (e.g. “this one”, “that”)
- ♦ Deducción por eliminación (e.g. “not that”, “other one”)

Resultados sobre el corpus GIVE

- ♦ Se está actualmente entrenando un árbol de decisión usando features contextuales
- ♦ El árbol entrenado será evaluado sobre 3 mundos diferentes



Resumiendo ...

- ♦ En un entorno virtual dado es posible entrenar un árbol de decisión para la GER
- ♦ El proceso de cómo intercalar modificación del mundo para simplificar las ER todavía es un tema abierto
- ♦ Pero es un área muy nueva en la que hay pocos recursos
- ♦ Por ahora estamos juntando recursos

Qué vamos a hacer hoy?

- ♦ Introducción a los entornos virtuales
- ♦ Herramientas para determinación del contenido
- ♦ Determinación del contenido como planning
- ♦ Estrategias de generación de referencias
 - ♦ El proyecto SCARE
 - ♦ El proyecto GIVE
- ♦ **Evaluación de sistemas de GLN**
 - ♦ El GIVE Challenge
- ♦ Conclusiones del curso

Evaluación de sistemas de GLN

- ♦ Evaluar sistemas de GLN es difícil:
 - ♦ Hay muchas formas de decir lo mismo, todas correctas
 - ♦ No hay un gold standard
- ♦ El trabajo de evaluación es muy reciente:
 - ♦ Sesión especial en la Conferencia Internacional en Generación de Lenguaje Natural (INLG) desde el 2006
 - ♦ Aparición de la organización “Challenges de generación” en el 2007
 - ♦ ASGRE-07 / REG-08 / TUNA / GREC Challenges
 - ♦ GIVE Challenge

REG/TUNA Challenge

- ♦ Paso 1: Anotadores human producen expresiones referenciales
- ♦ Paso 2: Cuán bien los sistemas de NLG reproducen las descripciones humanas?



Evaluando las métricas

- ♦ TUNA 08: La métrica automática de “humanidad” indicó que los sistemas eran más humanos que los humanos :P.
 - ♦ Sistemas automáticos tuvieron mejor nota que otros humanos
- ♦ Belz & Gatt 08 compararon la métrica de “humanidad” en TUNA contra métricas de utilidad de las ER para humanos (eg, tiempo de identificación de un objeto).
- ♦ No se encontró correlación entre las métrica automática de “humanidad” y las métricas de Belz y Gatt.

Evaluando las métricas

- ♦ Los puntos anteriores nos muestran los riesgos de definir métricas automáticas
- ♦ Es necesario seguir haciendo experimentos con humanos
- ♦ Las métricas con humanos son las más importantes, ya que generamos lenguaje natural para que lo lean las personas :-)

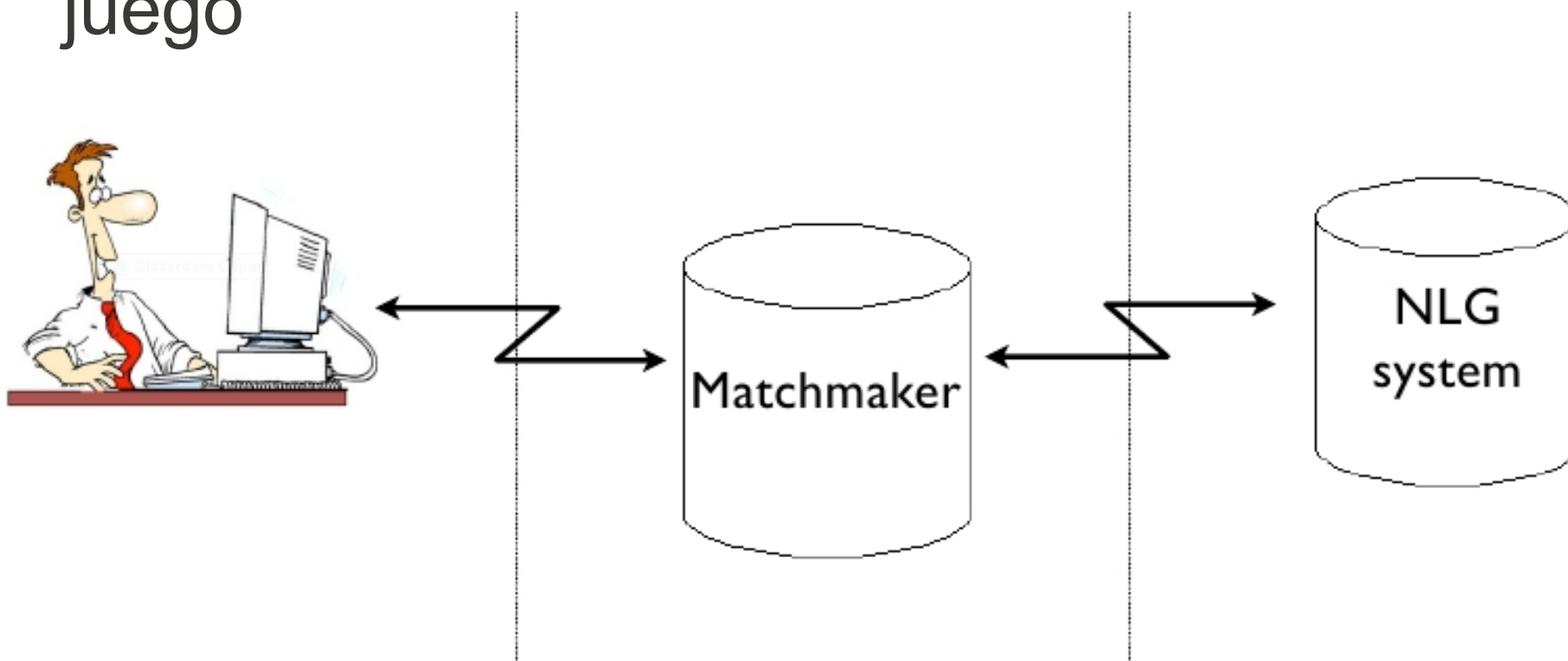
Problemas para evaluar GLN

- ♦ Dos opciones estándar:
 - ♦ Evaluar contra un gold standard: artificial para NLG porque hay muchas formas correctas de decir lo mismo
 - ♦ Evaluar usando anotadores o jueces humanos: más apropiado pero muy caro y lento

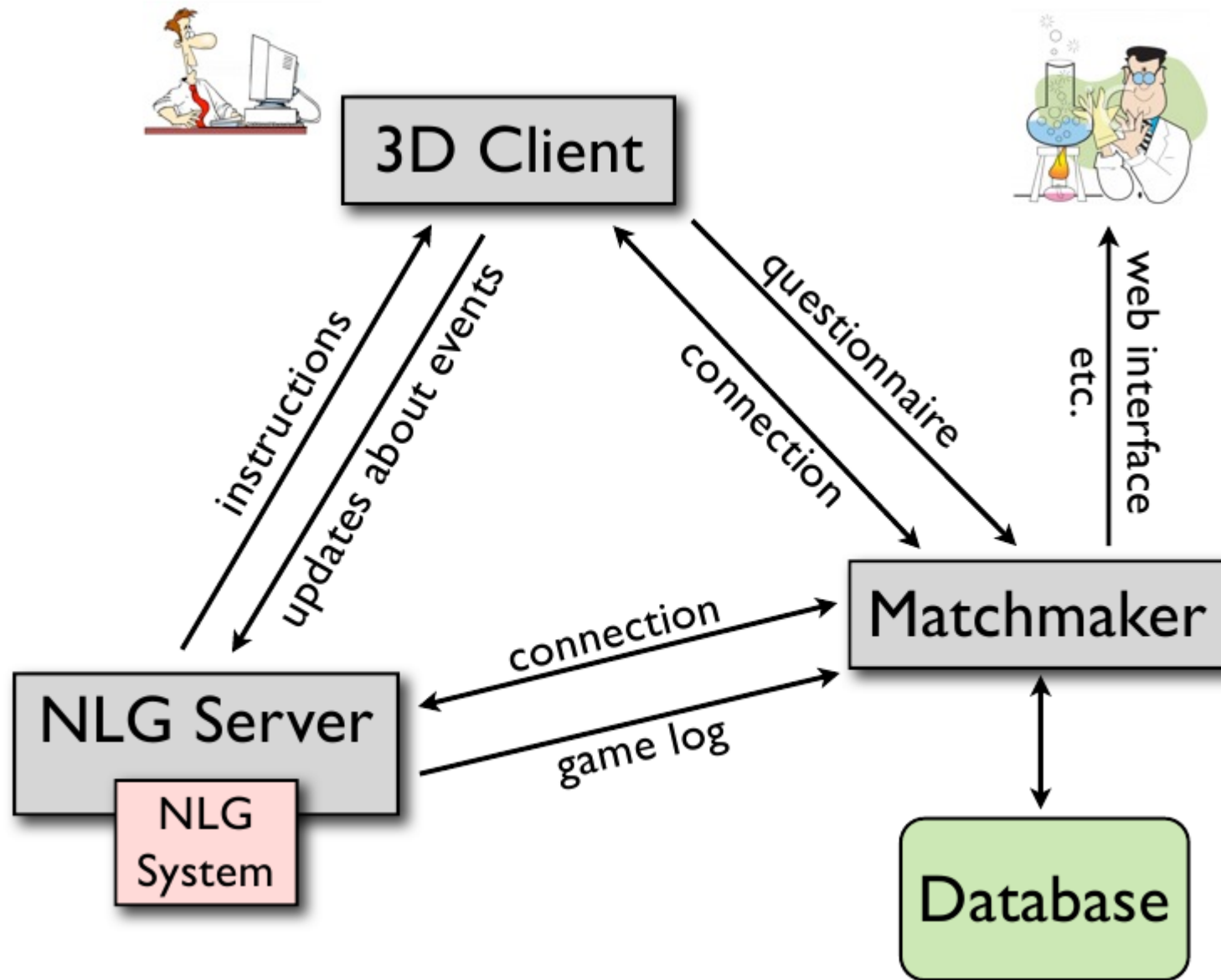
- ♦ Qué hacer?

Idea del GIVE Challenge

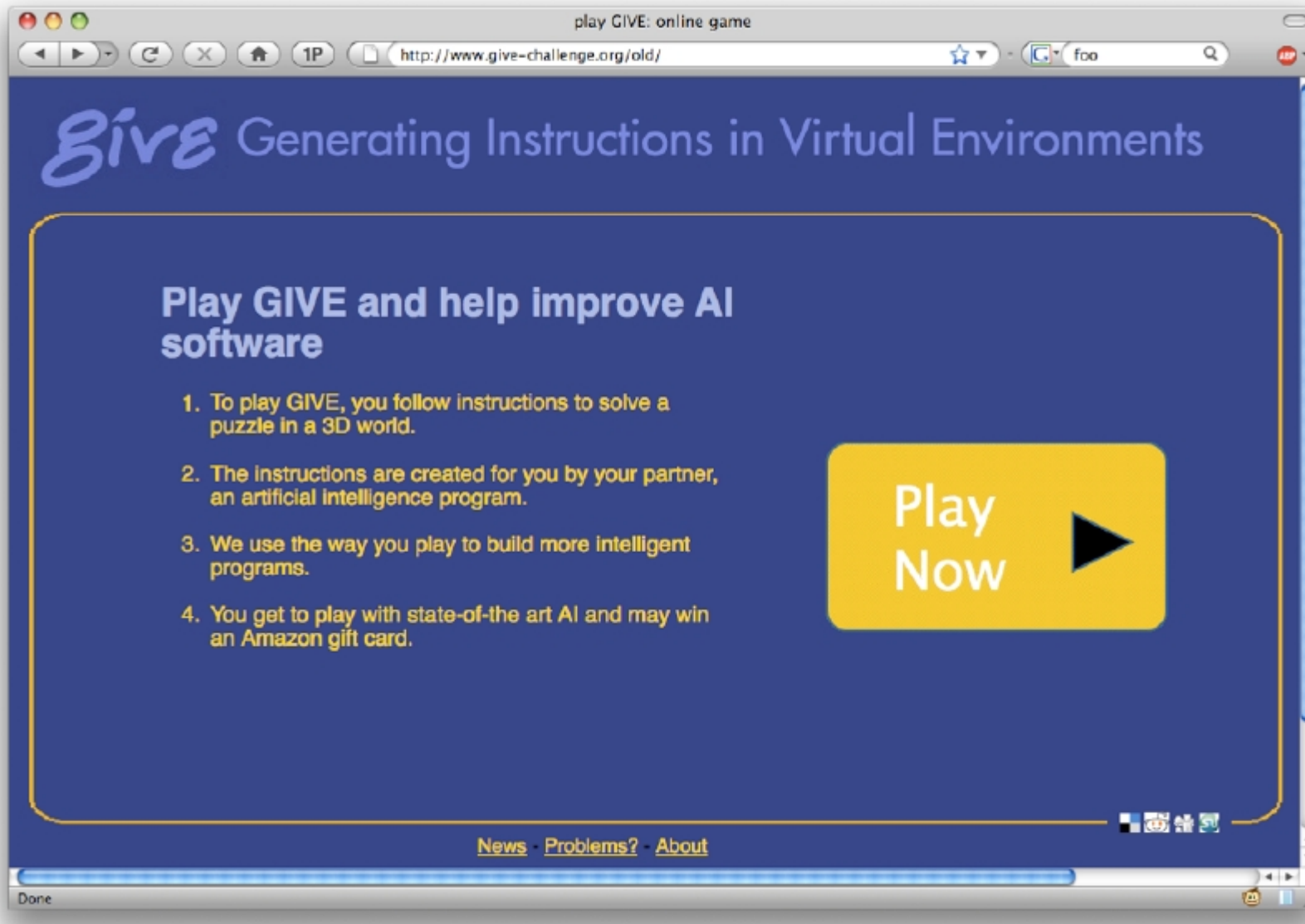
- ♦ Usar jueces humanos que no cobren!
- ♦ Hacer experimentos en internet usando un juego



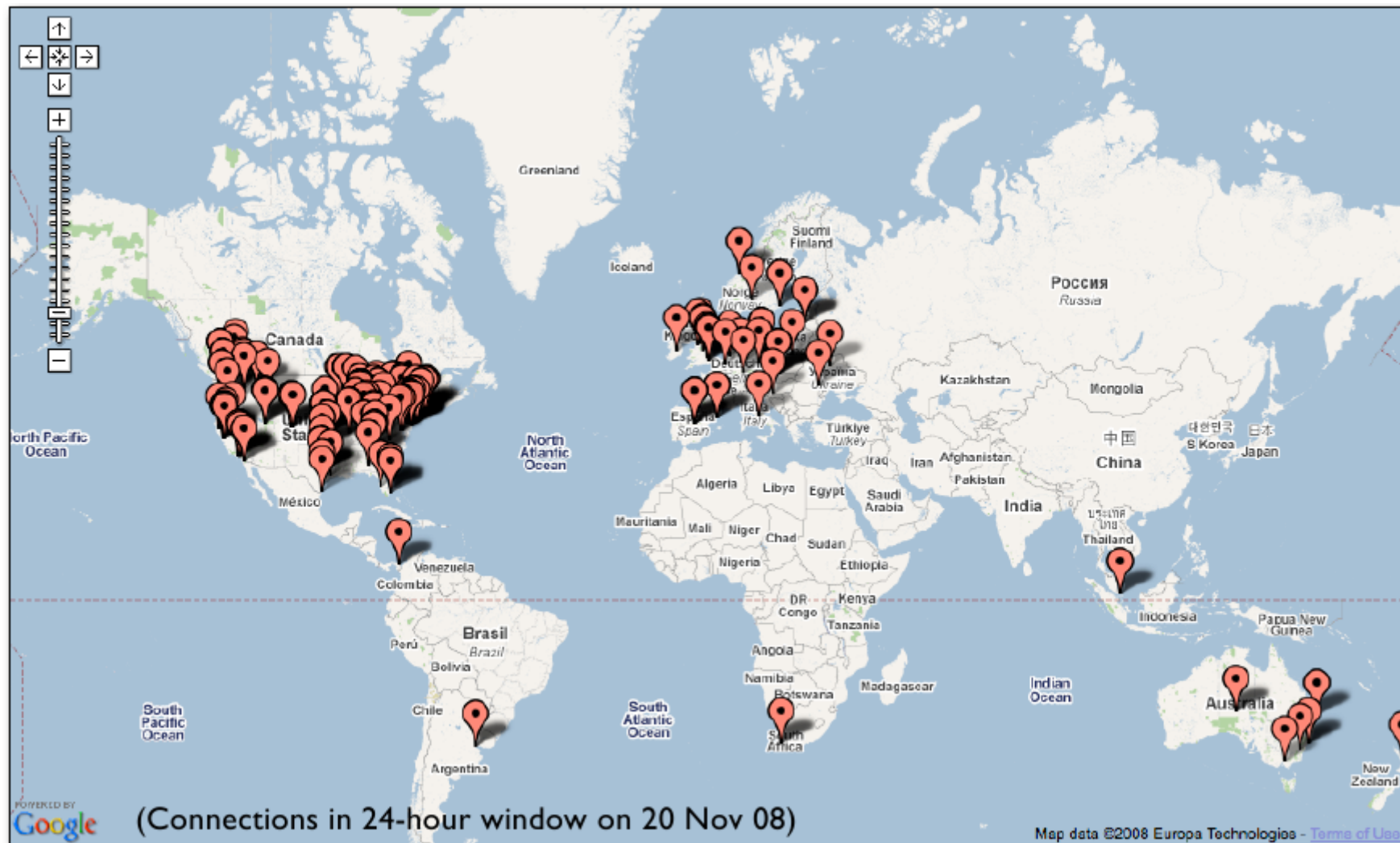
Infraestructura del GIVE Challenge



GIVE website



Recolección de datos



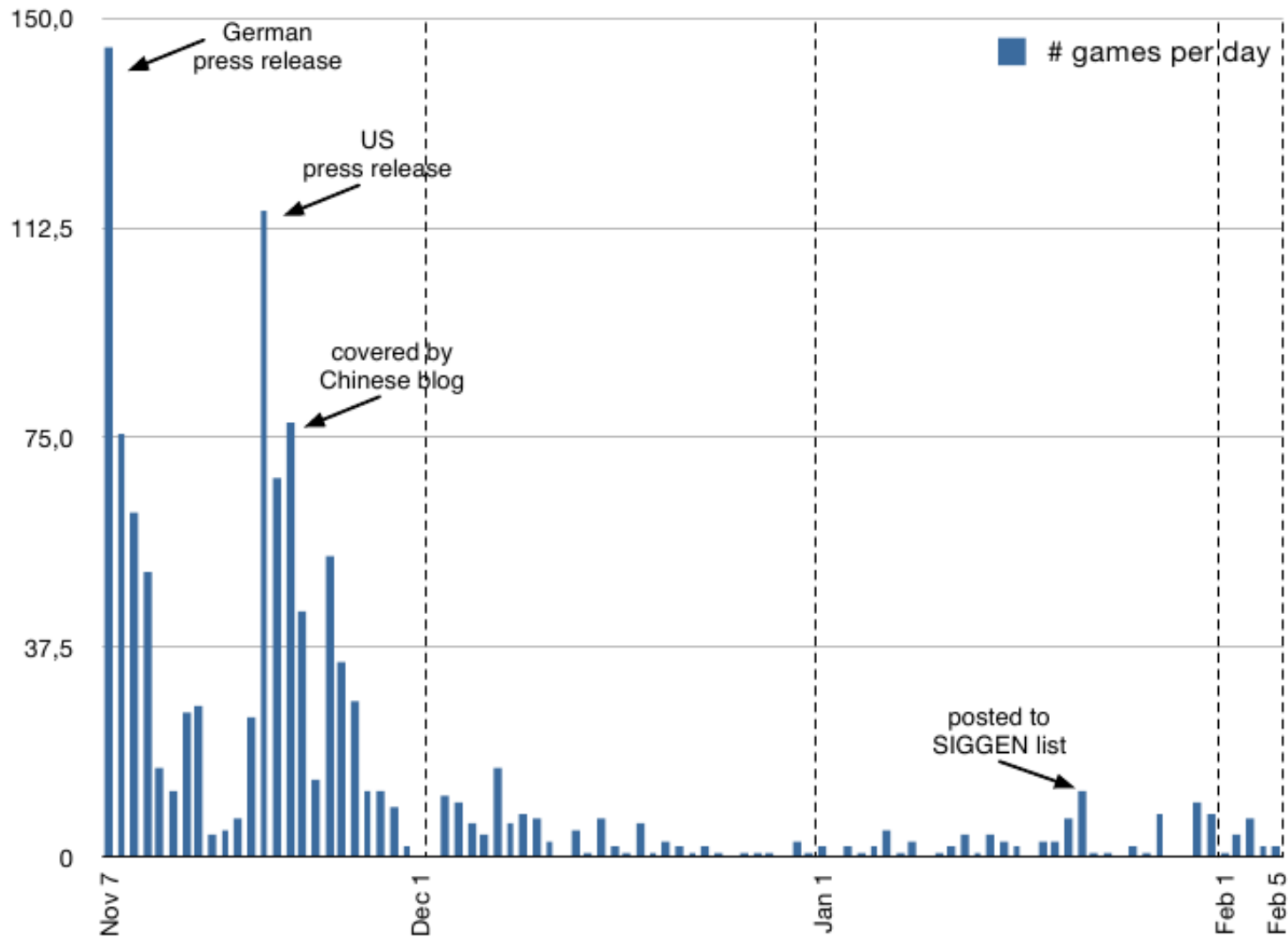
Recolección de datos

En GIVE 2009 se recolectaron más de 1000 juegos válidos en 3 meses

En GIVE 2010 se recolectaron más de 2000 juegos válidos en 2 meses y medio



Recolección de datos

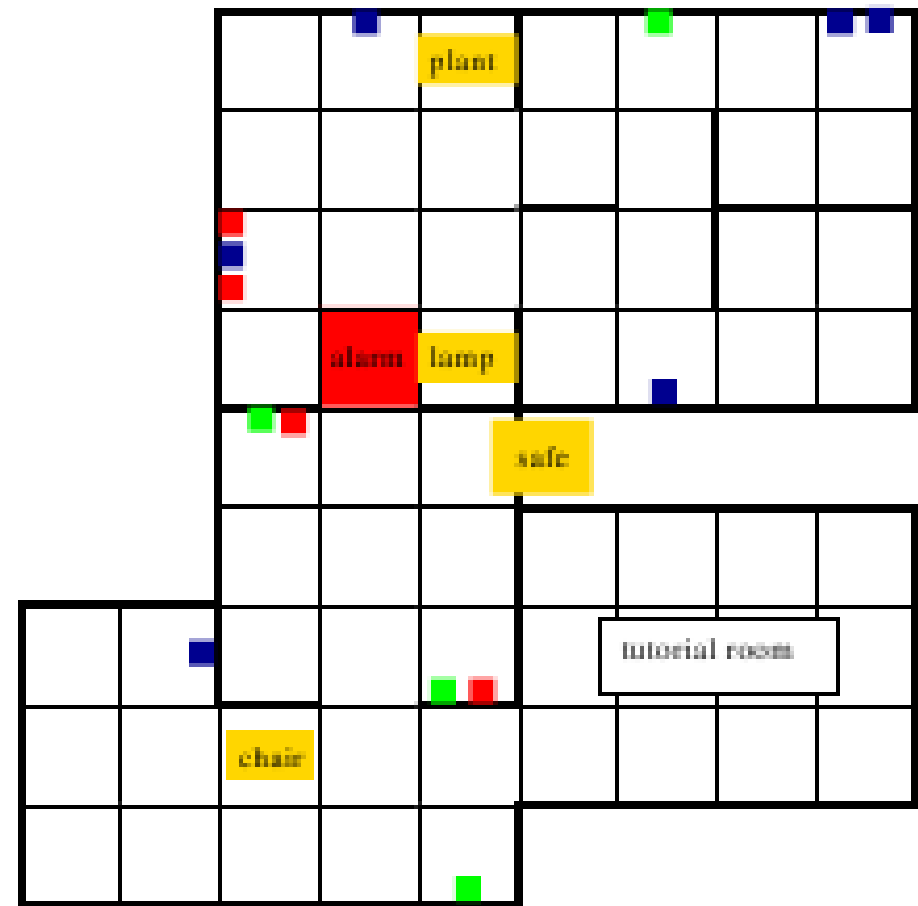


GIVE-1 y GIVE-2: Organización

- ♦ GIVE-1 y GIVE-2 fueron los esfuerzos de evaluación de sistemas de GLN más grandes que usaron datos de humanos.
- ♦ GIVE-1 evaluó 5 sistemas y GIVE-2 evaluó 7 sistemas que implementaban distintas estrategias
- ♦ Durante la evaluación los sistemas tuvieron resultados consistentes con resultados en el laboratorio
- ♦ El GIVE Challenge está respaldado por INLG, SIGSEM, SIGGEN de ACL

GIVE-1: Características

- ♦ La navegación en GIVE-1 era discreta, no continua.
- ♦ El mundo estaba dividido en baldosas y el usuario se podía mover una baldosa por vez
- ♦ “Girar” implicaba girar 90 grados por vez
- ♦ En un mundo había una cantidad finita y relativamente pequeña de posibles posiciones.



GIVE-1: Características



GIVE-1: Sistemas

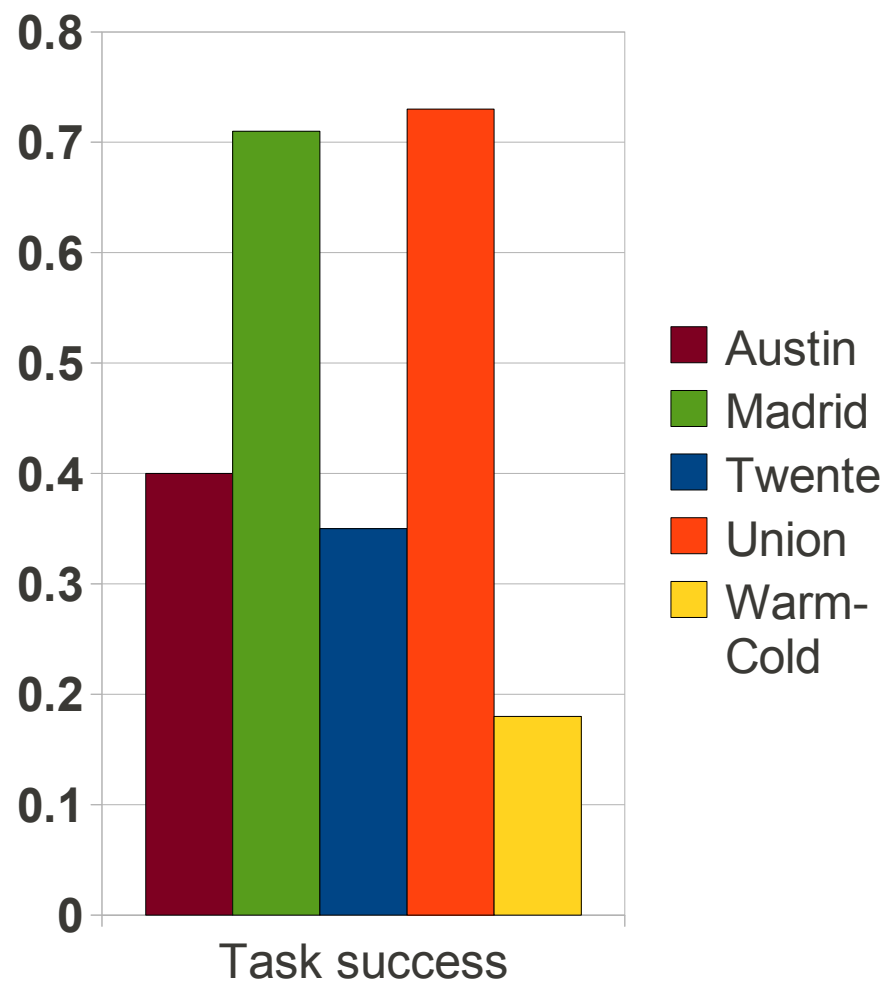
- ♦ **Austin (USA):** Baseline (verbalizar las acciones del plan una por una) más algo de agregación
- ♦ **Madrid (España):** Énfasis en inferir y explotar aspectos “ocultos” del mundo, como habitaciones, esquinas, etc.
- ♦ **Union College (USA):** Énfasis en instrucciones de navegación, switching entre modos basados en landmarks y modos basados en paths.
- ♦ **Twente (Holanda):** Énfasis en adaptación al usuario y a su habilidad de entender y seguir las instrucciones
- ♦ **Twente Warm/Cold system (Holanda):** Sólo dice “frío”, “tibio”, “caliente”; su objetivo era maximizar la diversión

GIVE-1: Métricas objetivas

1. Task success: El jugador alcanzó la meta?
2. Instrucciones: Número de instrucciones promedio por juego producidas por el sistema de GLN
3. Pasos: Número de acciones del jugador (manipulación más movimiento) promedio por juego
4. Acciones: Número de acciones de manipulación promedio por juego
5. Segundos: Tiempo en segundos promedio por juego

GIVE-1: Métricas objetivas

- ♦ A (baseline) logró el 40% del task success
- ♦ M y U, con estrategias diferentes, les fue muy bien
- ♦ T empeoró el baseline → implementar adaptación al usuario es difícil
- ♦ W le fue mal como era de esperarse



GIVE-1: Métricas objetivas

- ♦ M y U tuvieron **buenos** resultados en estas métricas
- ♦ T y W tuvieron **malos** resultados como era de esperarse
- ♦ Sin embargo, A tuvo **muy buenos** resultados!
- ♦ GIVE-1 fue muy fácil

	A	M	T	U	W
instructions	83.2 B	58.3 A	121.2 C	80.3 B	190.0 D
steps	103.6 A	124.3 B	160.9 C	117.5 A B	307.4 D
actions	11.2 B	8.7 A	14.3 C	9.0 A	14.3 C
seconds	129.3 A	174.8 B	207.0 C	175.2 B	312.2 D

GIVE-1: Métricas subjetivas

- ♦ A (baseline) tuvo resultados un poco mejores que M (ganador) en las métricas subjetivas
- ♦ Lo mismo para U
- ♦ Las métricas subjetivas son difíciles de evaluar (e.g. play again)

	A	M
task difficulty	4.3 A	4.3 A
goal clarity	4.0 A	3.7 A
play again	2.8 A	2.6 A
instruction clarity	4.0 A	3.6 A B
instruction helpfulness	3.8 A	3.9 A
informativity	46% B	68% A
overall	4.9 A	4.9 A

choice of words	4.2 A	3.8 B C
referring expressions	3.4 B	3.9 A
navigation instructions	4.6 A	4.0 B
timing	78% A	62% B
friendliness	3.4 A B	3.8 A

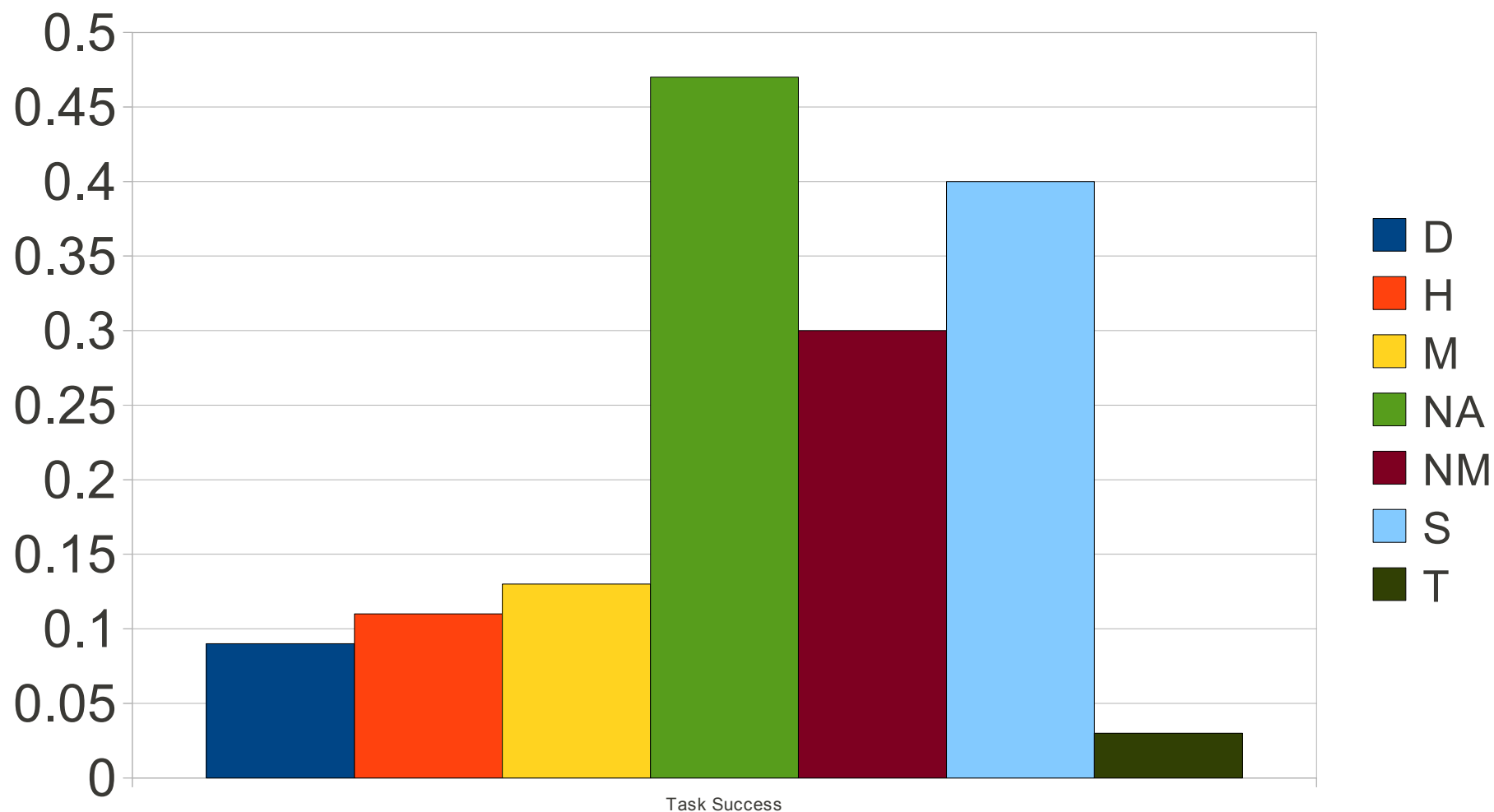
GIVE-2: Ideas para mejorar

- ♦ Problema: GIVE-1 fue demasiado fácil
- ♦ Solución propuesta: Que los mundos no sean discretos sino continuos:
 - ♦ El jugador se puede mover libremente
 - ♦ La cantidad de posiciones posibles son muchísimas (la posición y orientación del jugador usa nros reales)
- ♦ Problema: Las métricas subjetivas no fueron informativas
- ♦ Solución propuesta: Redefinirlas y clasificarlas, qué se está intentando medir?

GIVE-2: Sistemas

- ◆ Sistema D (Dublin Institute of Technology)
- ◆ Sistema T (Trinity College Dublin)
- ◆ Sistema M (Universidad Complutense de Madrid)
- ◆ Sistema H (University of Heidelberg)
- ◆ Sistema S (Saarland University)
- ◆ Sistema NA (INRIA Grand-Est in Nancy)
- ◆ Sistema NM (INRIA Grand-Est in Nancy)

Task Success en GIVE-2

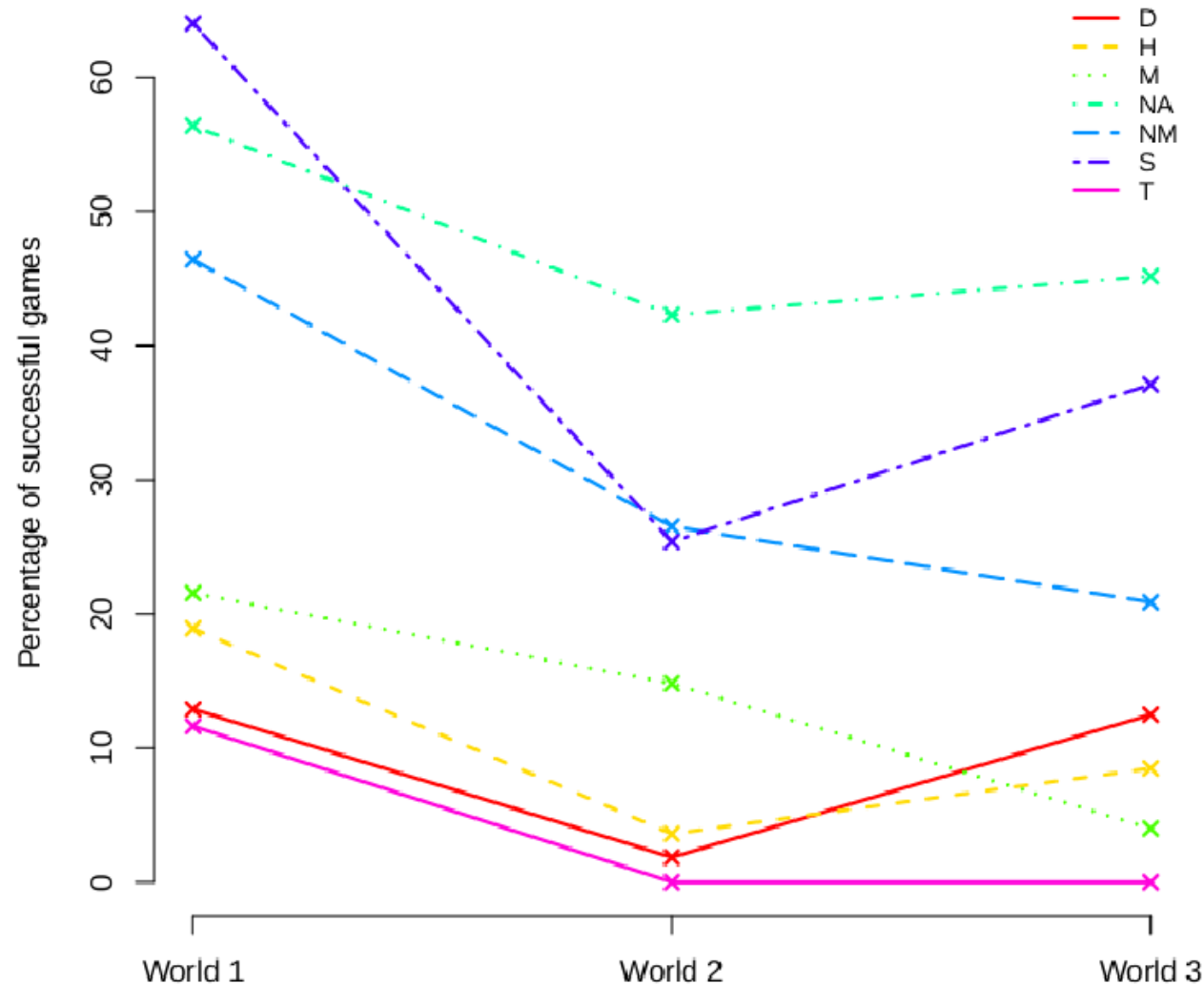


Resultados de GIVE-2

- ♦ GIVE-2 es mucho más difícil! El mejor sistema alcanza sólo el 47% de task success
- ♦ A sistemas que les fue muy bien en GIVE-1 (>70%) les fue muy mal en GIVE-2 (13%)
- ♦ Recuerden que NA usa granularidad L2 para content determination y usa deducción por eliminación para GER
- ♦ NM usa granularidad L3 y GER centradas en el usuario
- ♦ S usa granularidad L2 y conjuntos para referencia (e.g. press the leftmost button in the group of 2 red buttons)

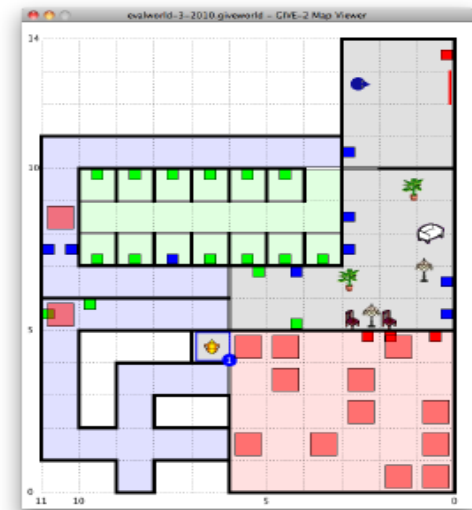
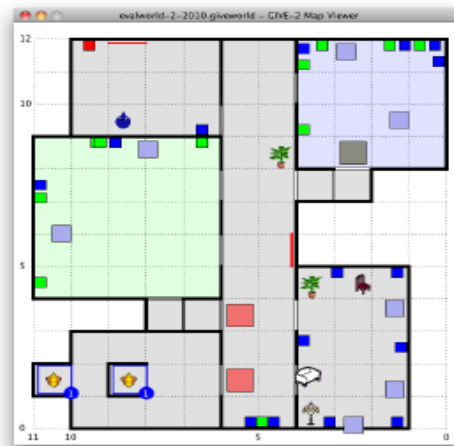
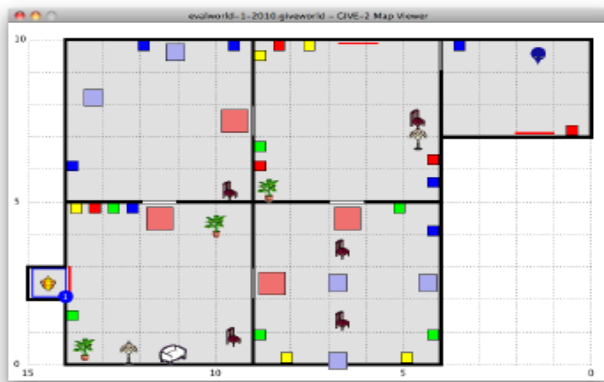
Efecto del mundo

- ◆ Algunos sistemas son mas robustos que otros al cambiar el mundo



Efecto del mundo

- ◆ El mundo 1 era el más fácil
- ◆ El mundo 2 tenía una grilla de botones y habitaciones alargadas
- ◆ El mundo 3 tenía muchas habitaciones iguales



Evaluación de sistemas de GLN

- ♦ Existen métricas automatizadas para calcular distancia de un texto gold estándar a un texto generado
 - ♦ Pero hay que usarlas con cuidado
- ♦ Es importante seguir evaluando con humanos
 - ♦ Pero es muy costoso
- ♦ La evaluación online ha probado ser un buen compromiso

Qué hicimos hoy?

- ♦ Introducción a los entornos virtuales
- ♦ Herramientas para determinación del contenido
- ♦ Determinación del contenido como planning
- ♦ Estrategias de generación de referencias
 - ♦ El proyecto SCARE
 - ♦ El proyecto GIVE
- ♦ Evaluación de sistemas de GLN
 - ♦ El GIVE Challenge
- ♦ Conclusiones del curso

Generación de lenguaje natural

- ♦ La generación de lenguaje natural es un área nueva y activa
- ♦ Los trabajos actuales van:
 - ♦ Desde sistemas ingenieriles ad-hoc para un trabajo específico
 - ♦ A estudio de temas teóricos como complejidad de gramáticas, algoritmos eficientes de planning, etc
- ♦ El uso de métodos estadísticos en el área aún es aislado y poco desarrollado

Generación en entornos virtuales

- ♦ No es lo mismo generar texto que generar texto durante una interacción
 - ♦ Cuando generamos texto no hay “segundas oportunidades”. No hay feedback ni oportunidad de corregirse
 - ♦ Esta observación, más un entorno dinámico (que cambia y puede ser cambiado) explican las principales diferencias entre ambas
- ♦ Ambas son difíciles de evaluar (cada una con sus pros y cons).
 - + Meta explícita, feedback continuo
 - Más variables que medir

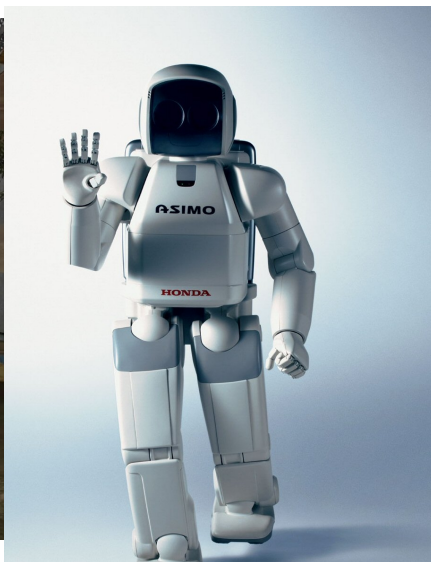
La comunidad de GLN

- ♦ SIGGEN (ACL Special interest group for Generation)
 - ♦ Website: <http://www.siggen.org/>
 - ♦ Recursos: software, papers, bibliografias
 - ♦ Anuncios de conferencias
 - ♦ Anuncios de posiciones y becas
 - ♦ Discusiones y gente en el área
 - ♦ Challenges: <http://www.nltg.brighton.ac.uk/research/genchal10/>
- ♦ La conferencias más importantes de GLN son INLG y ENLG (alternando una cada año)
- ♦ Papers GLN en ACL, EACL, IJCAI, AAI, SIGDIAL ...

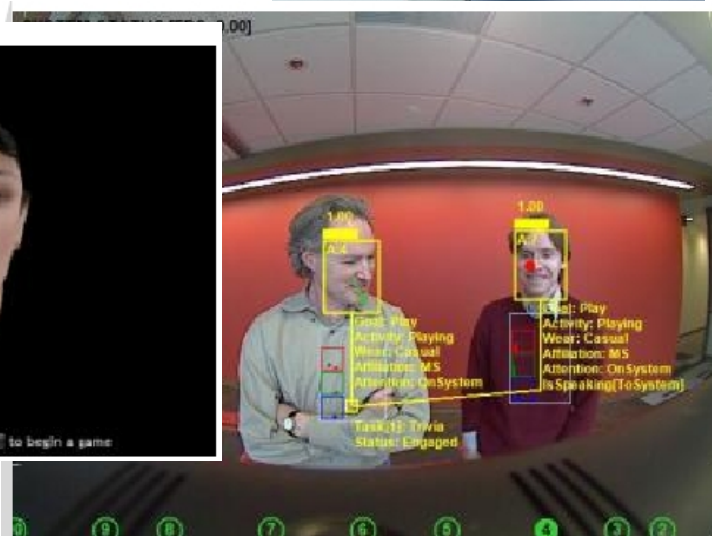
Gente en GLN

- ◆ Robert Dale@Macquarie University, AUS
- ◆ Ehud Reiter@Aberdeen University, UK
- ◆ Aravind Joshi@UPENN, USA
- ◆ Allan Black@MIT, USA
- ◆ Alexander Koller@Saarland University, GER
- ◆ Claire Gardent@INRIA Nancy Grand Est, FR
- ◆ Anja Belz@Brighton University, UK
- ◆ Albert Gatt@Malta University, Malta
- ◆ Laura Kallmayer@Tübingen University, GER
- ◆ Owen Rambow@Columbia University, USA
- ◆ ...

Empresas en GLN



- ♦ Honda Research (Mikio Nakano)
- ♦ Microsoft Research (Dan Bohus)
- ♦ Institute for Creative Technologies - USC (David Traum)
- ♦ Mitsubishi Electric Research Lab (Yves Schabes)
- ♦ Semantic Edge (Jörn Kreutel)



Recursos lingüísticos

♦ Gramáticas

♦ SEMFrag (Francés): http://led.loria.fr/en_ouils.php

♦ XMG-XTAG (Inglés):

<http://www.assembla.com/code/katya/subversion/nodes/XMG-basedXTAG>

♦ Surface realizers

♦ Geni: <http://trac.haskell.org/GenI/>

♦ Planners

♦ FF: <http://www.loria.fr/~hoffmanj/ff.html>

♦ SGPlan: <http://manip.crhc.uiuc.edu/programs/SGPlan/>

♦ TAG Parsers:

♦ SemConst: <http://sourcesup.cru.fr/xmg/SemConst/>

♦ Tulipa: <http://sourcesup.cru.fr/tulipa/>

Recursos de entornos virtuales

- ♦ SecondLife: <http://secondlife.com/>
 - ♦ Se puede programar dentro de SL usando Linden
 - ♦ Se puede conectar programas externos en cualquier lenguaje por puertos
 - ♦ Ojo: Tus creaciones dentro de SL no son tuyas
- ♦ OpenSim: <http://opensimulator.org/>
 - ♦ Entorno similar a SL pero open source
- ♦ Proyecto en SL:
 - ♦ Allegro: <http://www.allegro-project.eu>
 - ♦ The Virtual University of Edinbourg: <http://vue.ed.ac.uk/>
 - ♦ Y muchos más ... (si están interesados escribanme a luciana.benotti@gmail.com)

Oportunidades para estudiantes

- ♦ Masters Erasmus Mundus en Europa
 - ♦ Masters in Language Technologies (Francia, Alemania, Rep Checa, Malta, Italia, Holanda) <http://lct-master.org/> (contacto: Claire Gardent)
 - ♦ Masters in Computational Logic (Italia, Alemania, Austria, España, Portugal) <http://www.computational-logic.eu/> (contacto: Enrico Franconi)
- ♦ Pasantias en:
 - ♦ ICT-USC: <http://ict.usc.edu/> (contacto David Traum)
 - ♦ Microsoft (contacto Dan Bohus o Tim Paek)
 - ♦ AT&T (contacto Jason Williams)

EliC en ECI

- ♦ ELiC estuvo financiada en parte por NAACL (el 75% del subsidio fue a becas de estudiantes)
- ♦ Por otra parte estuvo financiada por ECI y por el grupo PLN de la Universidad de Córdoba: <http://www.cs.famaf.unc.edu.ar/~pln/>
- ♦ ELiC se organizará cada año en distintos lugares de Argentina.
- ♦ ELiC 2011 va a ser en Córdoba
- ♦ ELiC 2011 se organizará junto con WNLP 2011
- ♦ WNLP 2010 será junto con IBERAMIA:
<http://www.cs.famaf.unc.edu.ar/~laura/nlpw/>
- ♦ Estamos aplicando a fondos de NAACL para financiar el viaje de estudiantes al workshop de IBERAMIA