

Configuration Workshop 2015 Vienna



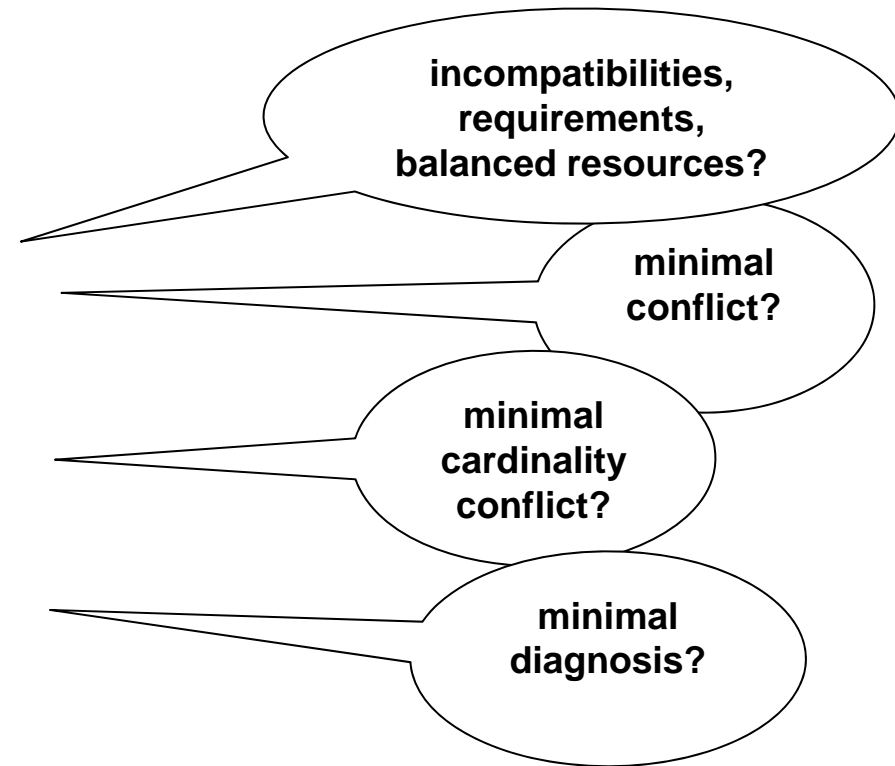
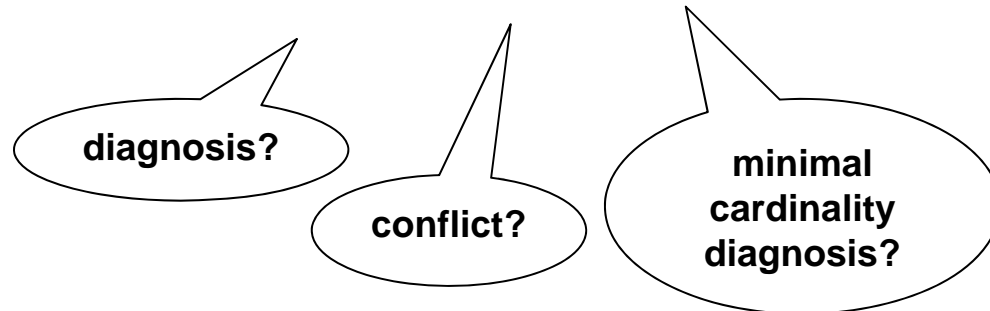
Learning Games for Configuration and Diagnosis Tasks

A. Felfernig, M. Jeran, T. Ruprechter,
A. Ziller, S. Reiterer, and M. Stettinger

Contact: alexander.felfernig@ist.tugraz.at

Learning Games: Motivation

students, domain experts, ...

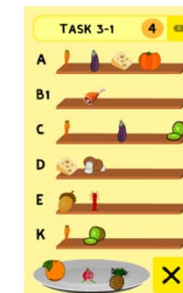
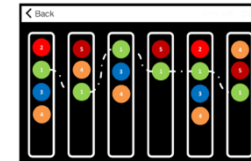


Challenge ➡ how to facilitate understanding?

Approach ➡ inclusion of „serious games“ into AI-related courses.

Agenda

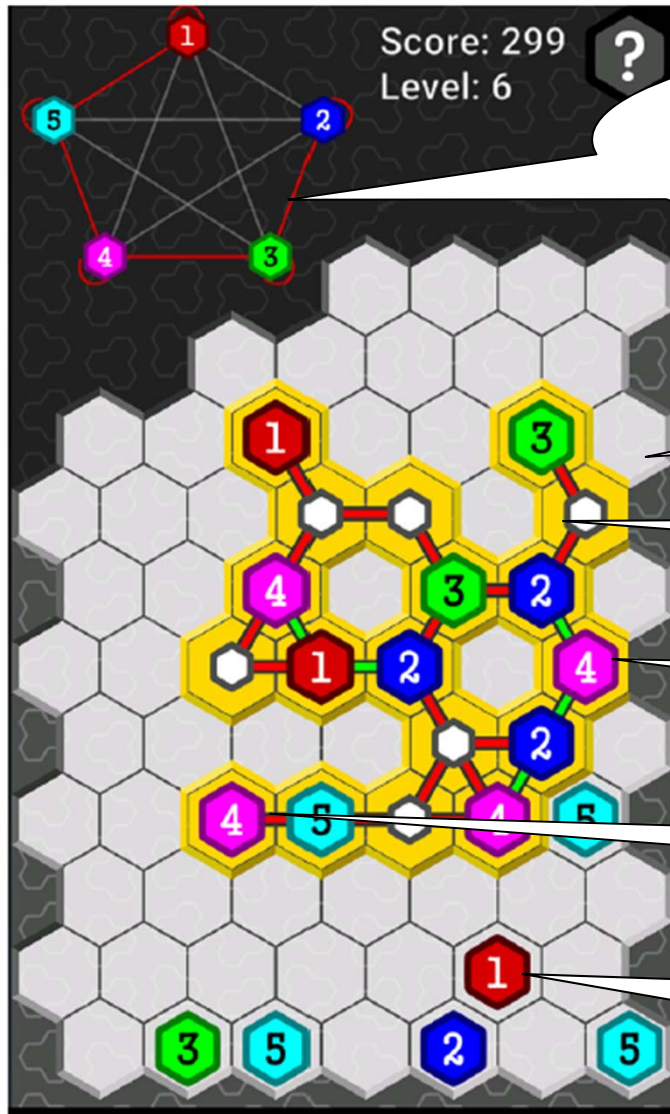
- Related Work
- Configuration Game
- Colorshooter and EatIt
- Configuration & Diagnosis Tasks
- User Study
- Future Work
- Conclusions



Related Work

- „Serious games“: entertainment not the primary goal, e.g., learning games
- 2-player map coloring: winner is the player who was last able to color a vertex consistently [Bodlaender 1991]
- General form: 2-player constraint satisfaction games [Börner et al. 2009]
- Generation of Sudoku puzzles using CSPs [Simonis 2005]
- Generation of crossword puzzles [Beacham et al. 2009]

Configuration Game: User Interface



edges in the graph specify constraints (incompatibilities) – e.g., two adjacent nodes (variables) in the honeycomb must not be instantiated with the combination „2-3“.

honeycomb structure, constraints, etc. are generated.

hexagons (nodes) in the honeycomb represent variables.

each variable has the same domain (in this case: 1-5)

incompatible assignments are indicated by red lines.

solution: each value is assigned to a variable and constraints are satisfied.

Configuration Task: Definition

Definition 1 (Configuration Task and Solution). A configuration task can be defined as a constraint satisfaction problem (V, D, C) where $V = \{v_1, v_2, \dots, v_n\}$ is a set of variables, $D = \cup \text{dom}(v_i)$ represents the corresponding domain definitions, and $C = \{c_1, c_2, \dots, c_m\}$ is a set of constraints. Additionally, user requirements are represented by a set of constraints $CREQ = \{r_1, r_2, \dots, r_k\}$. A solution for a configuration task is an assignment $S = \{inst(v_1), inst(v_2), \dots, inst(v_n)\}$ where $inst(v_i) \in \text{dom}(v_i)$ which is consistent with the constraints in $C \cup CREQ$.

Configuration Task: Example

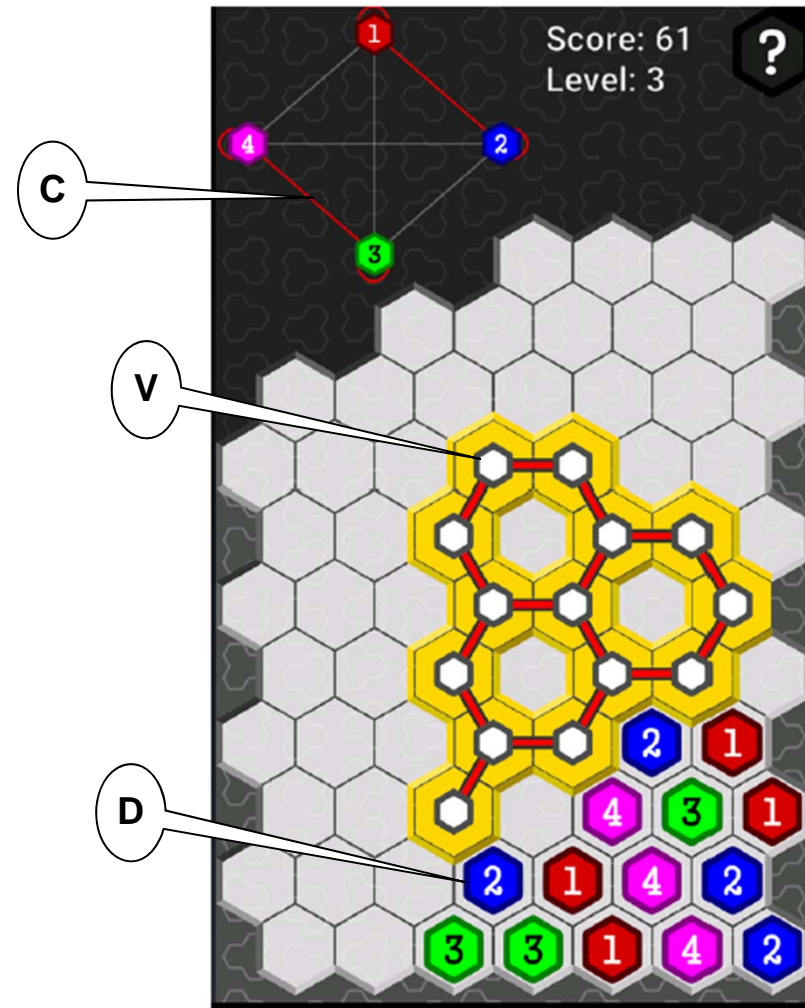
$$V = \{v_1, v_2, \dots, v_{14}\}$$

$$\text{dom}(v_i) = \{1, 2, 3, 4\}$$

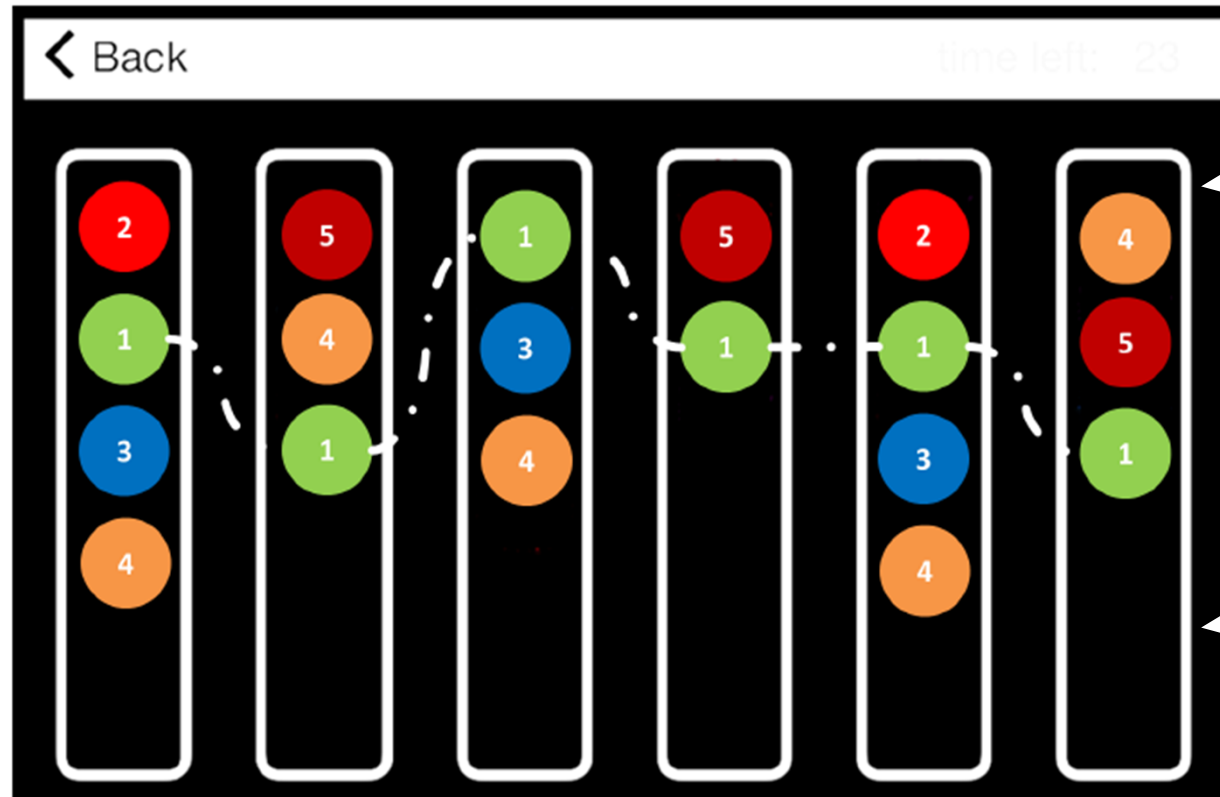
$$C = \{c_1, c_2, \dots, c_{16}\}$$

$$\begin{aligned}
 c_1 : & \neg(v_1 = 1 \wedge v_2 = 1) \wedge \\
 & \neg(v_1 = 2 \wedge v_2 = 2) \wedge \\
 & \neg(v_1 = 3 \wedge v_2 = 3) \wedge \\
 & \neg(v_1 = 4 \wedge v_2 = 4) \wedge \\
 & \neg(v_1 = 4 \wedge v_2 = 3) \wedge \\
 & \neg(v_1 = 3 \wedge v_2 = 4) \wedge \\
 & \neg(v_1 = 1 \wedge v_2 = 2) \wedge \\
 & \neg(v_1 = 2 \wedge v_2 = 1).
 \end{aligned}$$

$$CREQ = \{\}$$



ColorShooter: User Interface



goal: minimize the number of different colors needed to delete at least one item from each column.

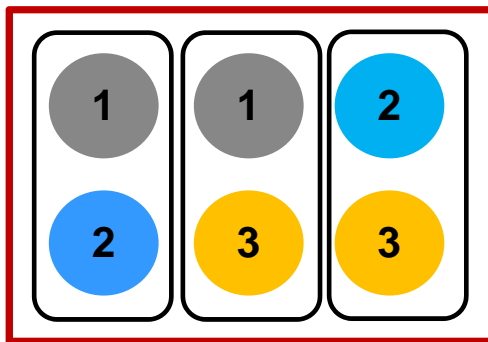
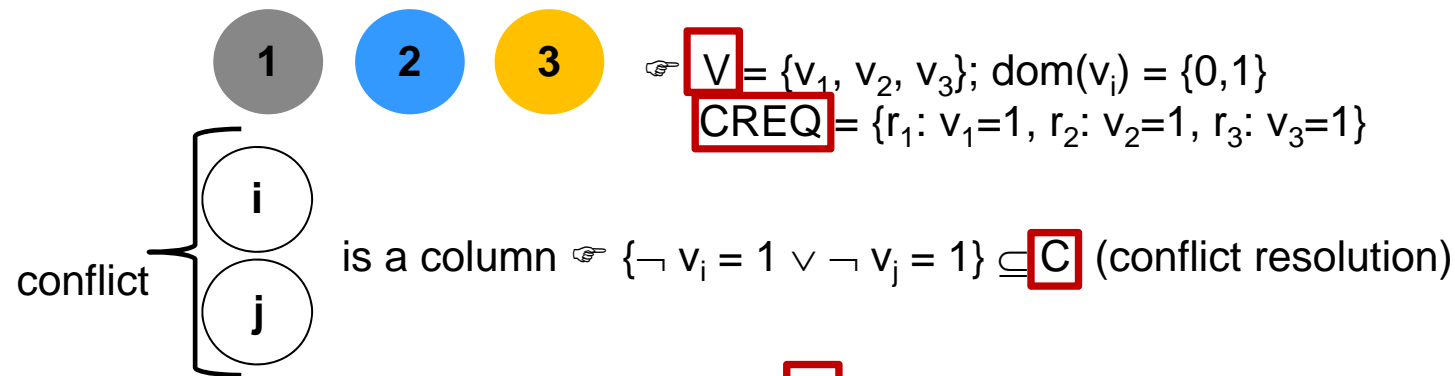
similar tasks: minimize decoys for potential fishes, ..., minimize meals for needed vitamins.

Each column represents a conflict that has to be resolved 🖱️ diagnosis task!

Diagnosis Task & Diagnosis

Definition 2 (Diagnosis Task and Diagnosis). A diagnosis task can be defined by a tuple $(C, CREQ)$ where C represents a set of constraints and $CREQ$ represents a set of customer requirements. If the requirements in $CREQ$ are inconsistent with the constraints in C , a diagnosis Δ ($\Delta \subseteq CREQ$) represents a set of requirements such that $CREQ - \Delta \cup C$ is consistent (in this context we assume that the constraints in C are consistent). Δ is minimal if $\neg \exists \Delta' : \Delta' \subset \Delta$.

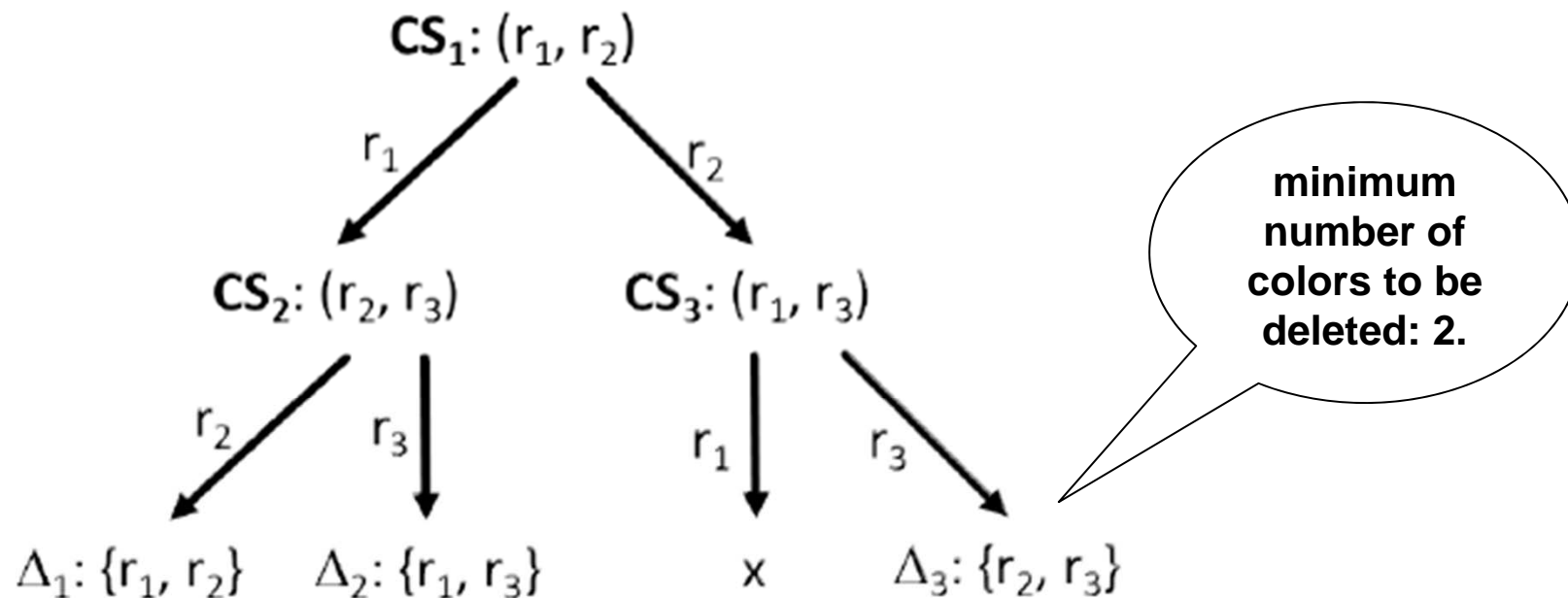
Diagnosis Task in ColorShooter



$$\begin{aligned}
 V &= \{v_1, v_2, v_3\} \\
 \text{dom}(v_1) &= \text{dom}(v_2) = \text{dom}(v_3) = \{0, 1\} \\
 C &= \{c_1 : \neg(v_1 = 1) \vee \neg(v_2 = 1), \\
 &\quad c_2 : \neg(v_1 = 1) \vee \neg(v_3 = 1), \\
 &\quad c_3 : \neg(v_2 = 1) \vee \neg(v_3 = 1)\} \\
 \text{CREQ} &= \{r_1 : v_1 = 1, r_2 : v_2 = 1, r_3 : v_3 = 1\} \\
 \underline{\Delta} &= \{r_1, r_2\}
 \end{aligned}$$

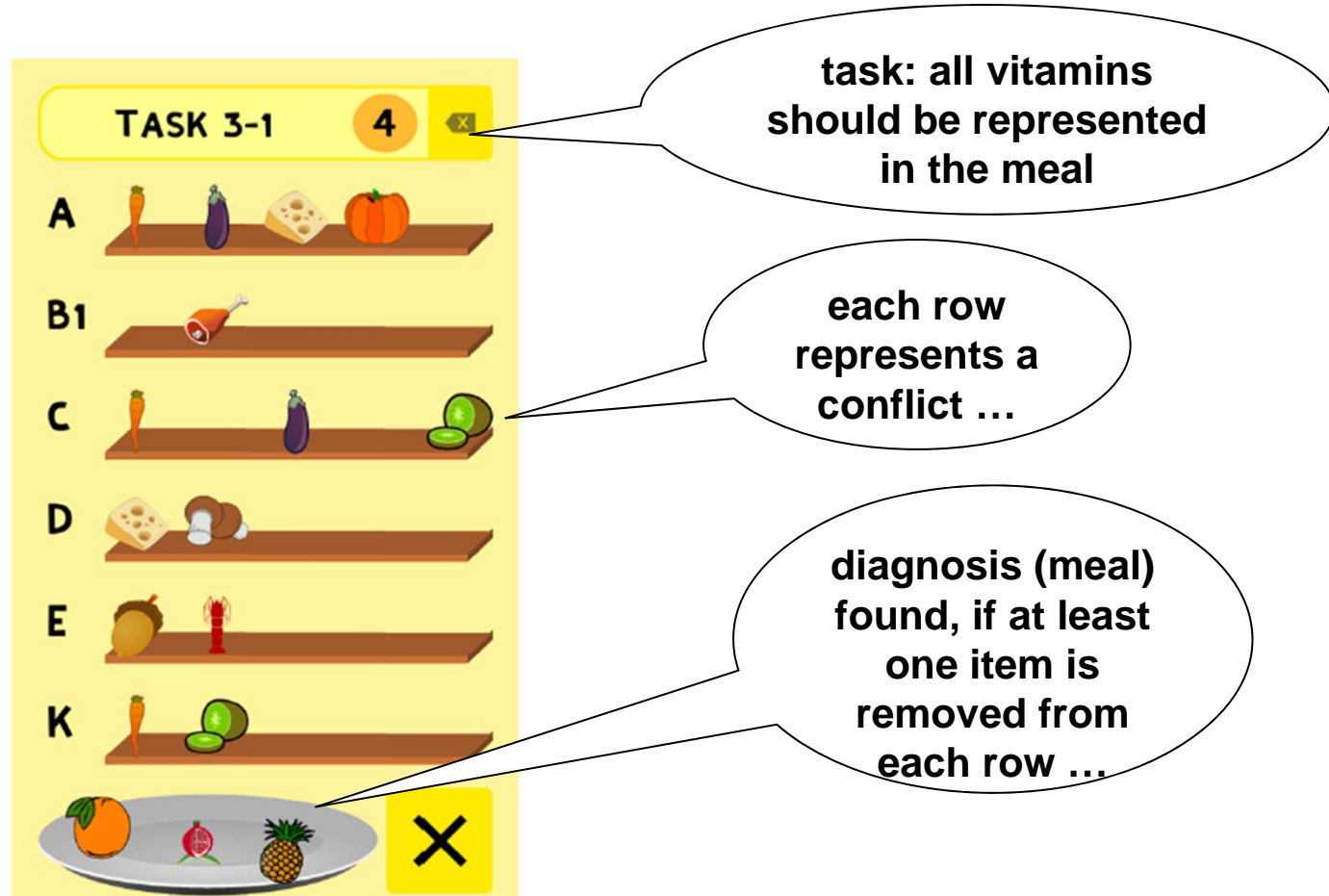
Conflict Sets: $\text{CS}_1 = \{r_1, r_2\}$, $\text{CS}_2 = \{r_2, r_3\}$, $\text{CS}_3 = \{r_1, r_3\}$

Diagnosis Determination: Basic Approach



Example hitting set directed acyclic graph derived from the conflict sets CS_1 , CS_2 , and CS_3 .

EatIt: Example



TASK 3-1 4

A carrot, eggplant, cheese, pumpkin

B1 chicken drumstick

C carrot, eggplant, green peas

D cheese, mushrooms

E acorn, chili pepper

K carrot, green peas

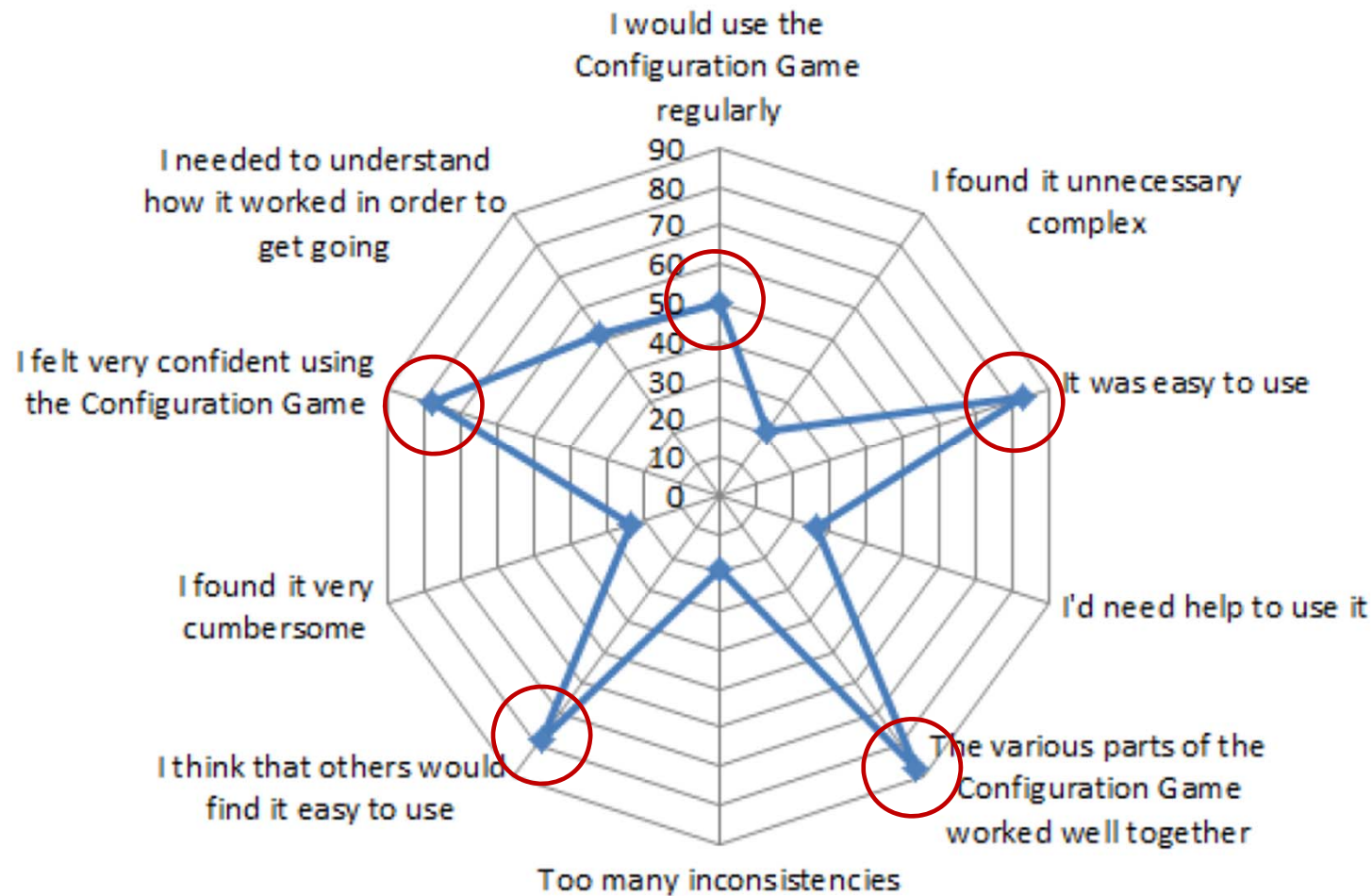
Meal: orange, radish, pineapple

task: all vitamins should be represented in the meal

each row represents a conflict ...

diagnosis (meal) found, if at least one item is removed from each row ...

Configuration Game: Usability Analysis



ColorShooter: Initial Impact Analysis

setting	all minimal diagnoses found (C_1)	all minimal diagnoses found (C_2)
played directly before	33%	20%
played one day before	20%	20%
did not play before	23%	10%

- two knowledge bases with two diagnosis tasks: C_1 , C_2
- in two settings students interacted with Colorshooter before solving diagnosis tasks
- $N = 60$ students

Future Work

- Improve usability of configuration game
- Include more complex constraint structures (ConfigurationGame)
- Analyze possibilities to further increase motivation to play the games
- Analyze potentials to improve knowledge acquisition interfaces based on experiences with games
- Implementation of further scenarios

Conclusions

- Some need to facilitate the understanding of AI concepts (e.g., diagnosis and configuration)
- „Serious Games“ can help to achieve this goal
- Initial prototypes have been developed (ConfigurationGame, Colorshooter, and EatIt)
- Results of initial studies indicate potentials to facilitate understanding of AI techniques

Thank You!

References

1. A. Beacham, X. Chen, J. Sillito, and P. vanBeek, 'Constraint Programming Lessons Learned from Crossword Puzzles', in AI 2001, LNAI, pp. 78–87. Springer, (2001).
2. H. Bodlaender, 'On the Complexity of Some Coloring Games', LNCS, 484, 30–40, (1991).
3. F. Börner, A. Bulatow, H. Chen, P. Jeavons, and A. Krokhin, 'The Complexity of Constraint Satisfaction Games and QCSP', Information and Computation, 207(9), 923–944, (2009).
4. H. Simonis, 'Sudoku as a constraint problem', in CP Workshop on Modeling and Reformulating Constraint Satisfaction Problems, pp. 13– 27, (2005).