Constraint modelling and solving: Learning from observing people

Ruth Hoffmann, Xu Zhu, Özgür Akgün, Miguel Nacenta, Peter Nightingale

ModRef 2022, Technion Haifa

What?

- Be concious of the human element of your applications and tools
- Be concious of how to explain/teach CP

Why?

- Gain knowledge into how people think about constraint problems and their solution
- Understand the potential and importance of appropriate representations of problems
- Improve understanding about how to build effective problem visualizations

Long term impact

- Benefit the designs of languages and interfaces for problem description
- Improve communication between the solver software and the end user
- Empower learners and practitioners of CP with a better understanding of important concepts

Limitations

- This is the first human-centred study addressing how people think about CP problems
- It is a small study
- It is an "unfocussed" study meant to inspire and inform

Setup – Participants

- 30 participants
- All either studying towards their first degree or with at least 1 degree
- Participants from 3 different backgrounds; non-CS, CS or CP
- 10 participants in each of the 3 groups

Setup – Problems

`https://github.com/stacs-cp/cp2022-hcicp`

Each participant was assigned two problems out of

1. Word Crypto
2. Subset Sum
3. Sudoku
4. Scheduling
5. Magic Square
6. Knapsack

Setup – Problems

`https://github.com/stacs-cp/cp2022-hcicp`

Word Crypto

Consider an addition:

```
    S E N D
  + M O R E
  ---------
  M O N E Y
```

Where each letter represents a number. Find a number for each letter which would satisfy the addition. Each letter represents a different number.

Setup – Problems

https://github.com/stacs-cp/cp2022-hcicp

Subset Sum

Given a set of numbers, find a subset of those numbers which sums up to zero. Find as many sets as possible.

Set: 1, -3, 5, 6, -2, 5, -7

Setup – Stages

- 3 stages, each approx 15mins
  1. Problem representation
  2. Problem solving
  3. Solver coding
- non-CS only participated in stages 1 & 2
- CS and CP participated in all stages

Problem Representation

How People Visually Represent Discrete Constraint Problems, X Zhu, MA Nacenta, Ö Akgün, P Nightingale (2019)

- The participants were asked to try to illustrate a problem to a friend assuming that they can only communicate using paper
- The participants were instructed to try to use as few words as possible in the specifications

Problem Solving

Understanding how people approach constraint modelling and solving, RH, X Zhu, Ö Akgün, MA Nacenta (2022)

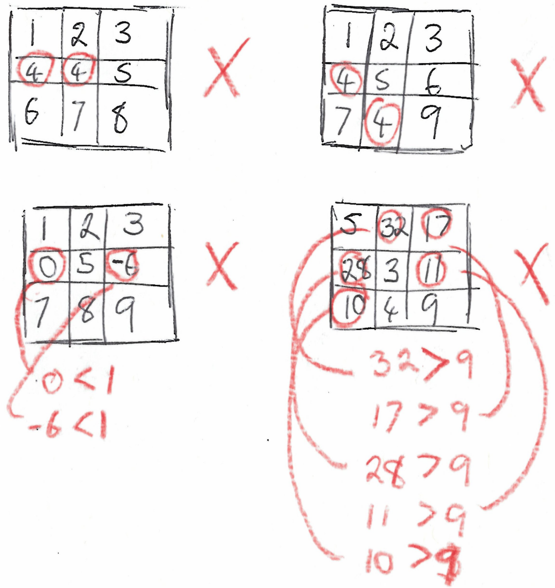
- The participants were asked to try to solve the problem by hand, without computer aid
- The participants were encouraged to talk aloud their thought processes and decisions
- Occasionally the experimenter would ask for clarifications or offered short reminders of the task

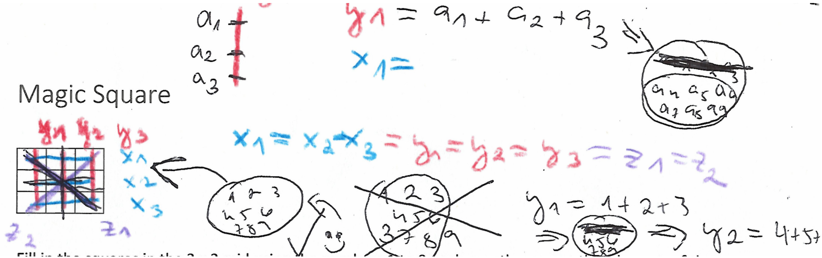
Solver Coding

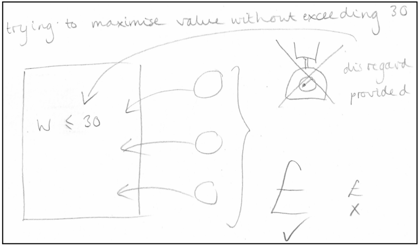
- The CS participants were asked to write pseudocode for an algorithm which can solve the tasks on paper
- We did not require the participants to produce a fast or optimal algorithm though they could do if they wished
- For CP participants, instead of pseudo-code, we asked them to write a constraints model using Essence' or another modelling language of their choice

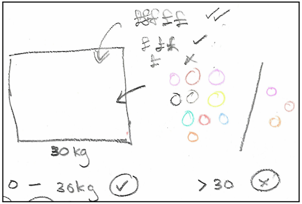
Representation

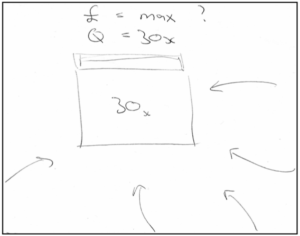
- Categorise different writings, drawings and doodles
- Analyse commonalities and differences

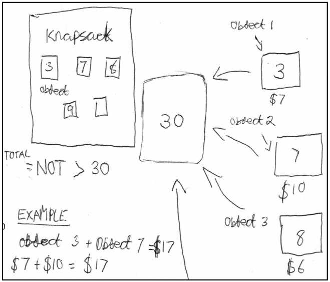

0<1
-6<1
32>9
17>9
28>9
11>9
10>9

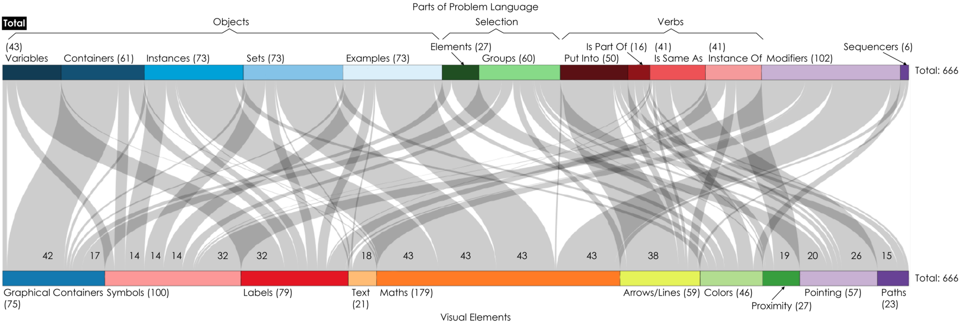

Magic Square
y1 = a1 + a2 + a3
x1 =
x1 = x2 = x3 = y1 = y2 = y3 = z1 = z2
y1 = 1 + 2 + 3
y2 = 4+5+


trying to maximise value without exceeding 30
disregard provided
W ≤ 30
£
£
X


30 kg
0 – 30kg
> 30

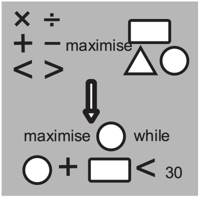

£ = max ?
Q = 30x
30x

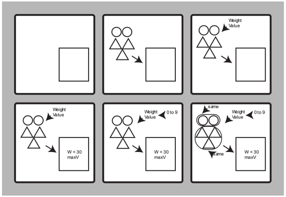

Knapsack
3 7 8
object
9 1
30
TOTAL
= NOT > 30
EXAMPLE
object 3 + object 7 = $17
$7 + $10 = $17
object 1
3
$7
object 2
7
$10
object 3
8
$6

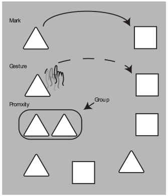

Parts of Problem Language
Total
Objects
Selection
Verbs
(43) Variables
Containers (61)
Instances (73)
Sets (73)
Examples (73)
Elements (27)
Groups (60)
Is Part Of (16)
Put Into (50)
(41) Is Same As
(41) Instance Of
Modifiers (102)
Sequencers (6)
Total: 666
42 17 14 14 14 32 32 18 43 43 43 43 38 19 20 26 15
Total: 666
Graphical Containers (75)
Symbols (100)
Labels (79)
Text (21)
Maths (179)
Arrows/Lines (59)
Colors (46)
Proximity (27)
Pointing (57)
Paths (23)
Visual Elements

Representation

1. Use a hybrid of math, text and visuals
2. Support the process
3. Leverage implicit information that the user produces

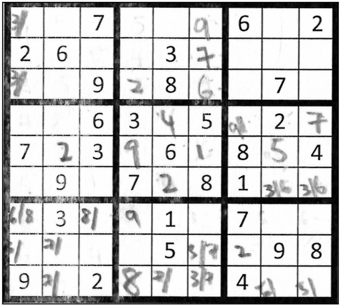


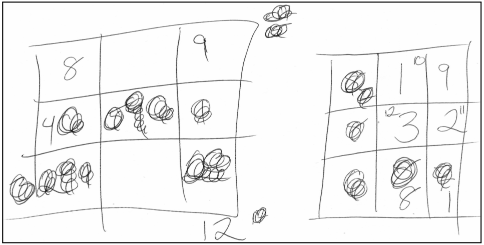


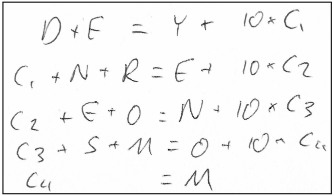


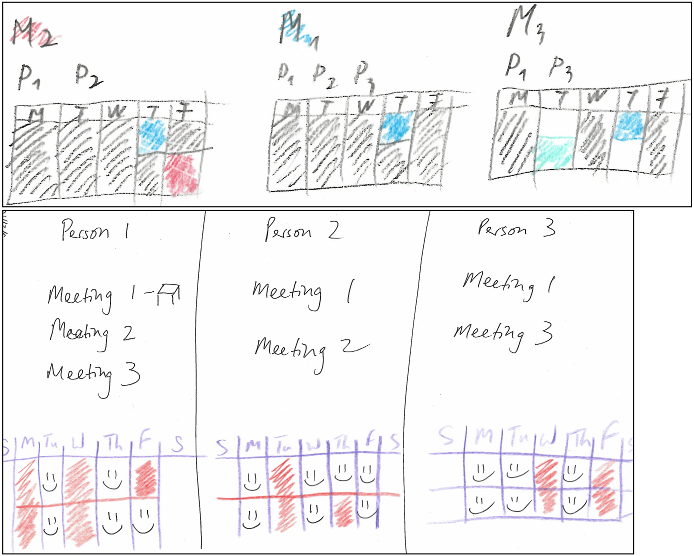
Solving

- Understand the human solving processes
- Categorise solving strategies
- Identify generally applicable strategies






D + E = Y + 10 × C1
C1 + N + R = E + 10 × C2
C2 + E + O = N + 10 × C3
C3 + S + M = O + 10 × C4
C4 = M


M2
P1 P2
M T W T F
M1
P1 P2 P3
M T W T F
M3
P1 P3
M T W T F
Person 1
Meeting 1
Meeting 2
Meeting 3
S M Tu W Th F S
Person 2
Meeting 1
Meeting 2
S M Tu W Th F S
Person 3
Meeting 1
Meeting 3
S M Tu W Th F

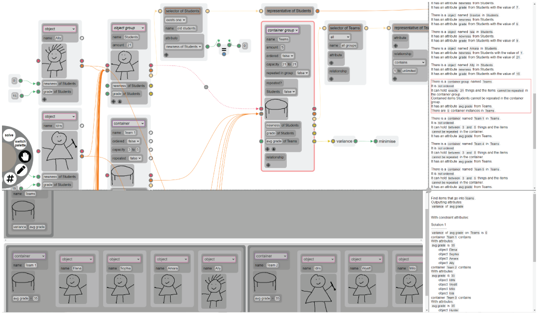
Solving

1. Non-experts are not aware of the implications of the problem representation
2. Visualisation of solving strategies has an impact on the understanding of a solving step
3. General belief that constraint problems are solved strategically

What have we learned?

We have an intuition for how

- to improve CP languages
- to visualise solvers
- people "learn" how to solve



Where to go next?

- Programming/modelling analysis
- Specialised studies
- Studies with a wider participant pool

What?

- Be concious of the human element of your applications and tools
- Be concious of how to explain/teach CP
- Inspect the human aspects of your work
- Think beyond your own experience
- Talk to us for collaborative studies

University of St Andrews | University of Victoria | University of York

Thank you!

rh347@st-andrews.ac.uk

xz32@st-andrews.ac.uk

ozgur.akgun@st-andrews.ac.uk

nacenta@uvic.ca

peter.nightingale@york.ac.uk