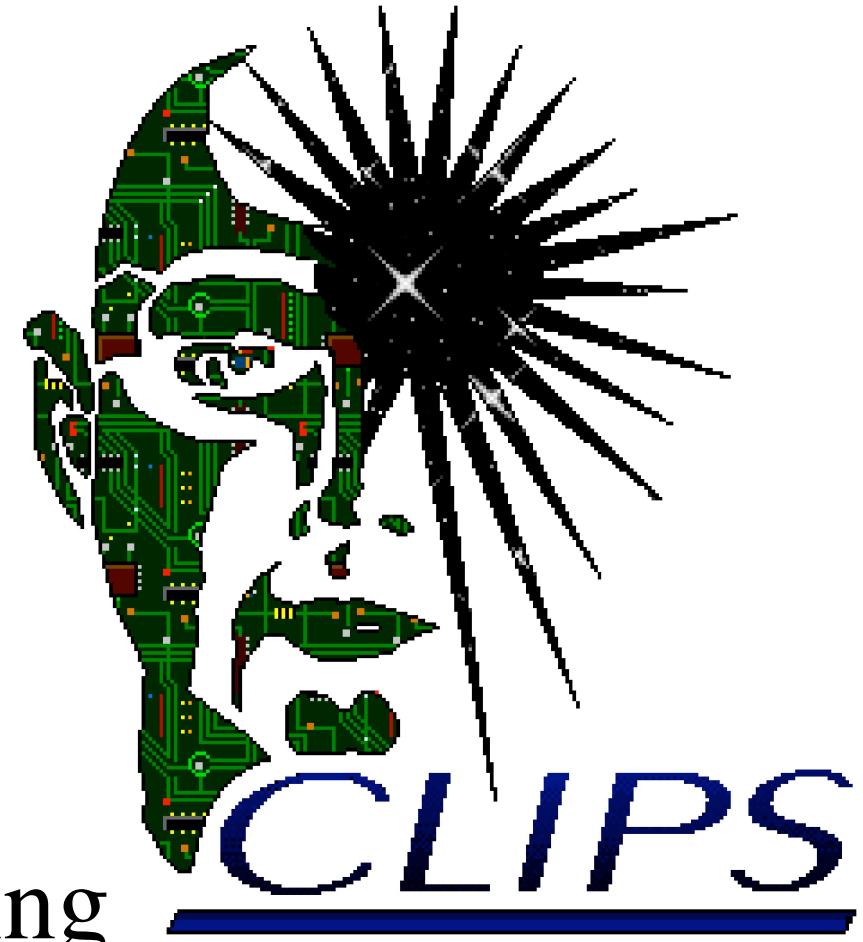
October Rulefest 2008



Rearchitecting

CLIPS

- C Language Integrated Production System
- Developed in 1986 by NASA at the Johnson Space Center
- Existing LISP expert system tools did not integrate well with the operational environment at NASA
- At the time, few expert system tools written in more conventional languages such as C
- Implementation in C made integration with languages such as C++, FORTRAN, and Ada easier
- While not a feature rich as commercial tools, CLIPS was ahead in recognizing that LISP was not a viable ecosystem for the widespread deployment of expert systems: a smaller, leaner tool could effectively compete in a number of niches

What's Happened in the Last Twenty Years

- The Winter of Artificial Intelligence
- Java
- The Internet
- Business Rules
- Open Source
- Faster Computers

C is Out, Java is In

- Open Source Java Engines: Drools, Hammurapi Rules, JEOPS, JRuleEngine, Mandarax, Open Lexicon, Prova, SweetRules, Take, TermWare, Zilonis, Jamocha, JLisa, OpenRules, Jrete, …
- Open Source .NET Engines: NxBRE, Drools.NET, SRE
- Open Source Perl Engines: INFeReNCZy, Perl Graphical Expert System Shell
- Open Source Python Engines: PyCLIPS, PyKE
- Open Source Ruby Engines: Ruby Rools, Ruleby
- Open Source Erlang Engines: ERESYE
- Open Source C Engines: CLIPS

Competing with Limited Resources

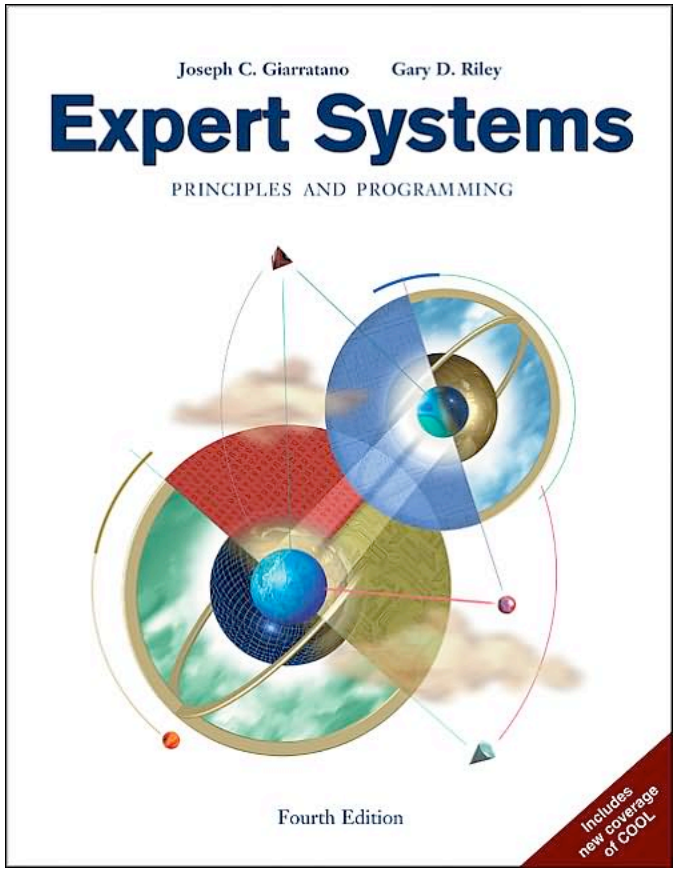
- Solve a problem you know
- Focus on specific niches, not general markets
- Focus on usability not features
- Focus on common user issues
  - Internationalization
  - Integration
  - GUI Development
  - Examples
- Focus on strengths
  - Speed/Footprint
  - Complex problems/Programmers
  - Academic users

Integration

- The amount of knowledge needed by users to integrate systems is often a barrier to their use
- Don't make multiple users duplicate effort for common issues
- Provide wrappers (Java/C++) that simplify access to the CLIPS API
- Provide a bridge to GUI development (Swing, Eclipse)
- Provide a bridge to other functionality (Excel, Database, Arrays, Hash Tables)
- Allow objects from other languages to be manipulated by CLIPS—Requires reference counts and other hooks
- Allow other languages access to expert system technology
- Dynamic memory tracking
- Contextual information

Internationalization

- *Expert Systems: Principles and Programming* originally published in 1989
- Fourth edition published in 2005
- Translations in Spanish, Chinese, and Russian
- How to support these users?
- UTF-8
- Swing Console



The Need for Speed

- How important is speed?
- Is speed a differentiating factor between tools?
- Is speed an advantage of the C language?
- Are cross-product benchmarks meaningful?
- What benchmark exists?
- Are existing benchmarks valid?
- Are existing benchmarks misleading?
- What benchmarks are needed?
- What changes enhance performance in CLIPS 6.3?

My First Benchmark

- Monkey and Bananas
- A simple planning problem in which a monkey must perform a sequence of actions in order to retrieve and eat a bunch of bananas
- Extended an existing OPS5 program
- Consists of 32 rules requiring 81 rule firings and a maximum of 32 facts in working memory in order for the monkey to complete its goal
- Unarguably a toy problem and could be better solved with a combination of forward and backward chaining
- Nonetheless it was quite useful for internal performance testing and gauging best case performance

Monkey and Bananas

Performance Then (Circa 1986)

Tool	Platform	Rules/Second
ART	Symbolics	72
OPS5	VAX 11/780	62.3
CLIPS	VAX 11/780	16.8
OPS5+	IBM AT	15.8
CLIPS	HP 9000	6.5
OPS5+	Macintosh	5.8
CLIPS	IBM AT	4.3
ExperOPS5	Macintosh	1.5

Monkey and Bananas

Performance Today

Tool	Platform	Rules/Second
CLIPS 6.3	2 GHz MacBook	100,500
CLIPS 6.24	2 GHz MacBook	118,900

Why Speed?

- For most, an incremental improvement. For some, a significant improvement
- A natural strength for a C language application
- Knowing it can be done makes the job much easier
- It’s an interesting problem
- Counter the misperceptions caused by existing benchmarks

Manners and Waltz Benchmarks

- The main benchmarks used for cross-product comparisons
- Developed decades ago
- Controversial for a number of reasons
- There is general agreement that these benchmarks are inadequate
- No replacements have been put forward
- In spite of these issues, the frequency with which these benchmarks are cited gives them credence

Who's Right?

"These benchmark programs test the ability of rule based systems to deal with large working memories and large numbers of full and partial matches for the rules. Benchmarks of this nature are interesting because they are the best predictors of how well a rule interpreter can handle complex rule bases."

"Nevertheless, if you omit the tricks, Miss Manners is a very good measure of how fast a rules-inferencing system will run on any give platform and CPU. The Waltz benchmark is another old test, but a good one."

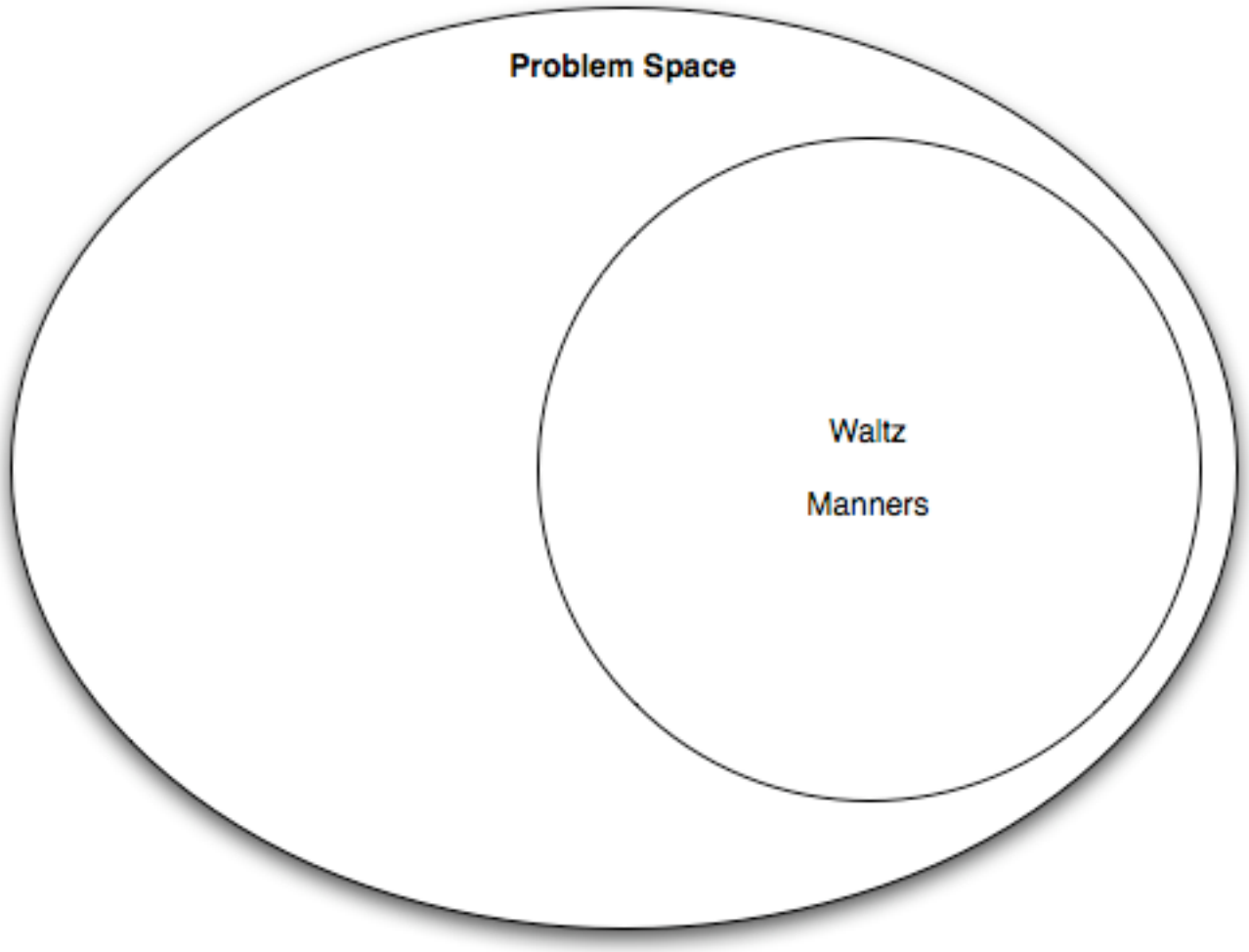
"The issues with the academic benchmarks for BRMS are fairly well documented, and there seems to be a general consensus that they should not be used for reliable product performance comparison, however they keep cropping up!"

"Beware these benchmarks!"

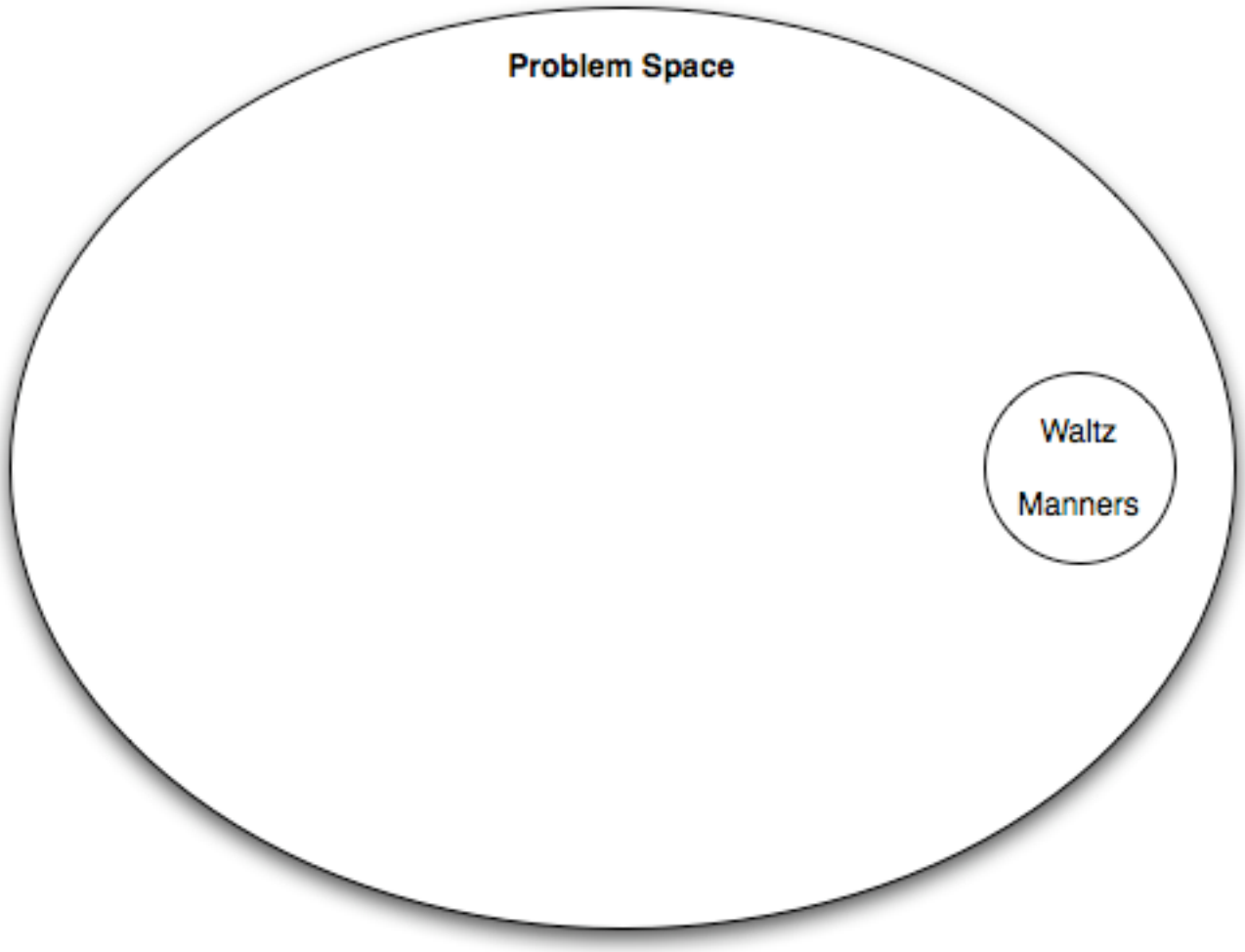
Perception is Reality

	Manners	Waltz
Product A	2,600	13,800
Product B	14	190

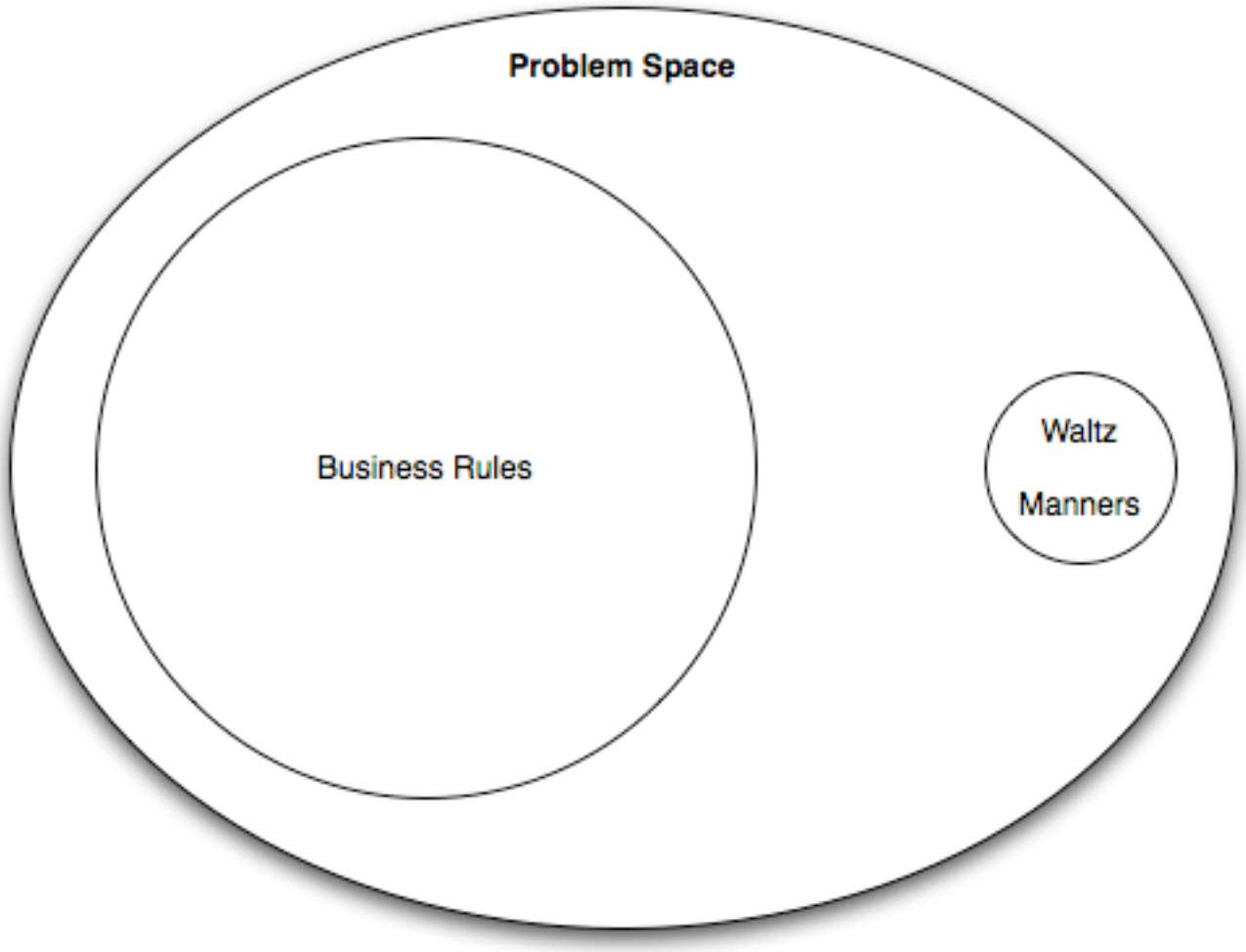
Waltz and Manners are Typical



Waltz and Manners are Atypical



Waltz and Manners are Irrelevant



Manners Overview

- Finds an acceptable seating arrangement for guests at a dinner party, seating everyone next to a member of the opposite sex with a similar hobby
- http://www.jboss.org/drools/documentation.html
- Constraint Satisfaction Problem
- Consists of eight Rules
- Data is scalable (8, 16, 32, 64, 128…)
- Problem domain is well understood

Waltz Overview

- Processes a collection of lines and labels them according to a set of constraints
- Consists of 32 rules
- Data is scalable (12, 25, 37, 50)
- Problem domain is more esoteric

Manners/Waltz Benchmark Results

	Manners 128	Waltz 50
CLIPS 6.3	2,600	13,800
CLIPS 6.24	14	190

Key Changes for Performance Improvement

- Added salience groups to improve performance with large numbers of activations of different saliences
- Hashed alpha memories
- Hashed beta memories
- Dynamic resizing of fact hash table.
- Constant selector in fact pattern nodes
- Lazy computation of ‘not’ conditional element count
- Asymmetric retract: partial matches have parent/child links
- Cleanup of partial match data structures
- Nested rete topology
- Removed addition of initial-fact patterns

Problems with Manners

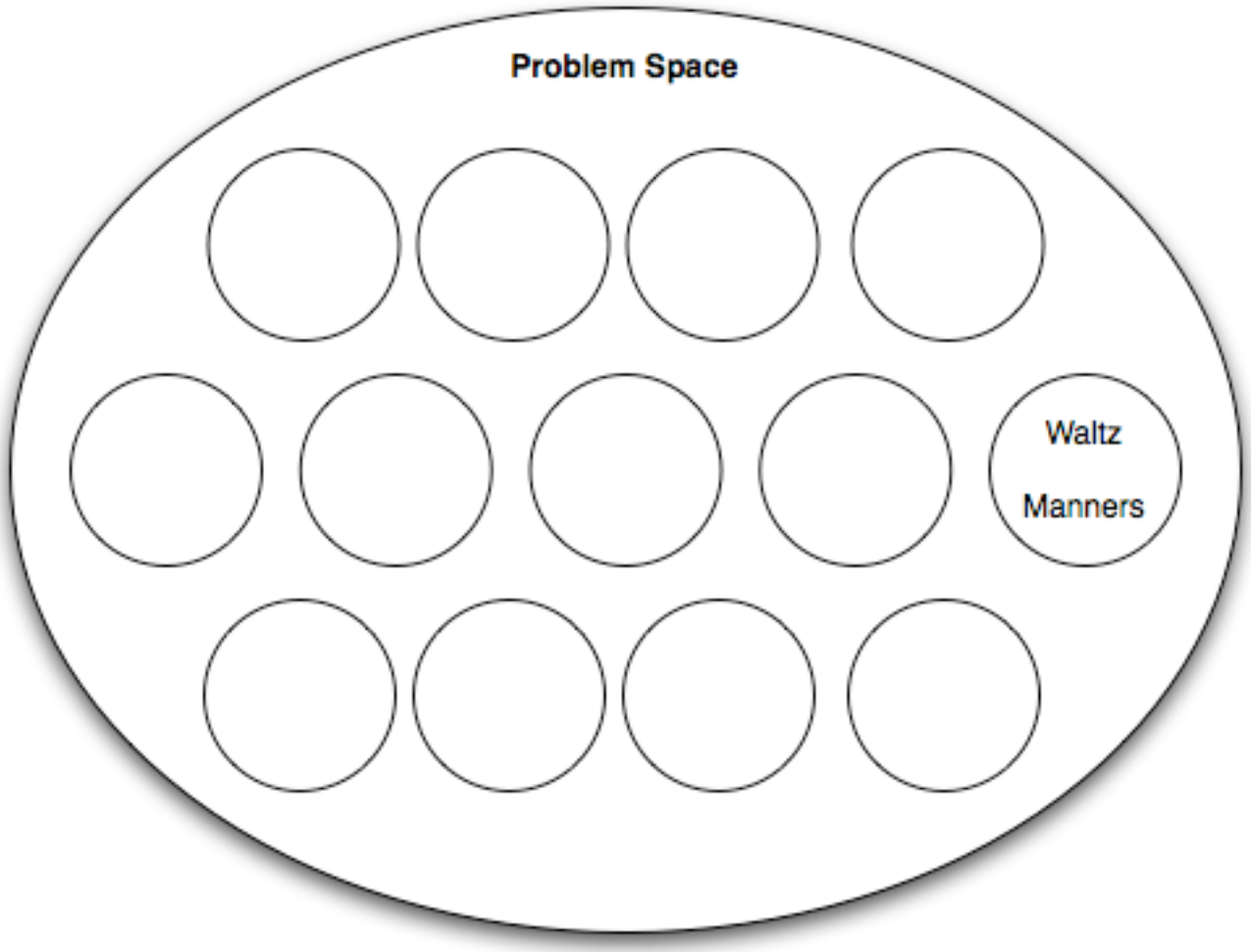
- Poorly written
- Small number of rules with a single rule determining performance
- Multiple solutions (although same number of rule firings)
- Targeted toward a specific algorithm
- Not representative of a rule application
- Performance comparison requires identical implementation
- Requires OPS5 compatible conflict resolution strategy
- Guilt by association (if a tool doesn't implement hashing assumes overall poor performance)
- Easy to cheat

Rewriting Manners

- Reorder the patterns in the find_seating rule, moving the count pattern to the end
- Use modules (i.e. multiple agendas) rather than context patterns to reduce churning of the partial matches

	Original	Reorder	Modules	Both
CLIPS 6.3	2,600	3,800	4,200	13,800
CLIPS 6.24	14	27	27	720

We Need More Benchmarks



Why Most Real World Applications Don't Make Good Benchmarks

- Problem domain is not well understood
- Non-portable features
- Procedural code
- Inconsistent results
- Unavailable
- Translation is time-consuming

Event Benchmark

- Derived from an existing system with over 5,000 rules
- First benchmark contains 703 rules
- All contain a single pattern matching two to five constant slot values
- 676 facts asserted causing 676 rule firings
- No chaining involved

	Rules/Second
CLIPS 6.3	222,200
CLIPS 6.24	111,100

Sudoku Benchmark

- Solves sudoku puzzles and lists the techniques used
- Consists of 80 rules
- Implements 18 different solution techniques
- Test case for unique rectangles technique:

```
The puzzle is:

   * * 4  * 6 2  8 * *
   * 8 *  * 1 *  6 2 *
   * * *  4 * *  * * *

   3 * *  * * *  2 * *
   9 7 *  * * *  * 3 8
   * * 1  * * *  * * 7

   * * *  * * 7  * * *
   * 2 3  * 9 *  * 8 *
   * * 6  2 3 *  1 * *
```

Sudoku Example Solution

The solution is:

```
1 3 4  7 6 2  8 5 9
5 8 7  3 1 9  6 2 4
2 6 9  4 5 8  7 1 3

3 4 8  9 7 5  2 6 1
9 7 2  1 4 6  5 3 8
6 5 1  8 2 3  9 4 7

4 1 5  6 8 7  3 9 2
7 2 3  5 9 1  4 8 6
8 9 6  2 3 4  1 7 5
```

Rules used:

- Naked Single
- Hidden Single
- Locked Candidate Single Line
- Locked Candidate Multiple Lines
- Naked Pairs
- Hidden Pairs
- Naked Triples
- Hidden Triples
- Swordfish
- Color Conjugate Pairs
- Multi Color Type 2
- Forced Chain Convergence
- Forced Chain XY
- Unique Rectangle

Sudoku Benchmark Advantages

- Designed to fire the same number of rules regardless of the conflict resolution strategy for any engine that supports salience/priority values
- Solutions are unique and easily verified
- There are test cases for each of the 18 supported solution techniques. The design also allows the benchmark to be incrementally developed and tested as the more complex solution techniques are not employed until the easier ones fail.

Sudoku Benchmark Advantages

- Although it's a bit more complex than manners, the problem domain is easily understood and there's a lot of information online describing the various solution techniques used
- Like manners and waltz, the amount of data can be scaled by selecting the size of the puzzle (3x3 x 3x3, 4x4 x 4x4, ...).
- It can be run in non-stressed and stressed modes to demonstrate how implementation techniques can effect performance.

Sudoku Unique Rectangles Benchmark Statistics

	Rules Fired	Mean Activations	Maximum Activations	Mean Facts	Maximum Facts
Non-Stress	7,986	855	2,184	855	2,184
Stress	8,343	1,935	5,686	878	2,185

	Run 6.24	Run 6.3	Rules/Second 6.24	Rules/Second 6.3
Non-Stress	6.9 s	0.77 s	1,158	10,439
Stress	27.1 s	1.7 s	300	5,038

SICM Benchmark

- Shipping Instruction Comparison Module
- A CLIPS-based expert system that automatically processes changes to cargo container shipping instructions.
- In production use since 2005 and has gone through several revisions to include more functionality.
- Benchmarked version contains 462 rules. Current version contains 634 rules.
- Data set size can range up to 10,000 or more facts.

SICM Benchmark Results

Benchmark	Rules Fired	Mean Activations	Maximum Activations	Mean Facts	Maximum Facts
2	993	397	2,303	4,280	4,518
5	2,843	7,828	42,014	2,314	3,264
6	9,135	579	657	4,587	4,634
7	4,801	2,174	6,686	5,207	5,657

Benchmark	Load 6.24	Load 6.3	Run 6.24	Run 6.3	Rules/Second 6.24	Rules/Second 6.3
2	2.1 s	2.1 s	0.47 s	0.34 s	2,117	2,886
5	0.8 s	0.8 s	3.00 s	1.22 s	947	2,334
6	2.8 s	2.8 s	0.36 s	0.20 s	25,445	44,779
7	3.0 s	3.2 s	1.27 s	0.92 s	3,792	5,207

All Benchmark Results

	Rules	Rules/Second6.24	Rules/Second 6.3
MAB	32	118,900	100,500
Events	703	111,100	222,200
Sudoku	80	1158	10,439
SICM	462	947	2,334
Sudoku Stress	80	300	5,038
Waltz 50	32	190	13,800
Manners 128	8	14	2,600

Questions