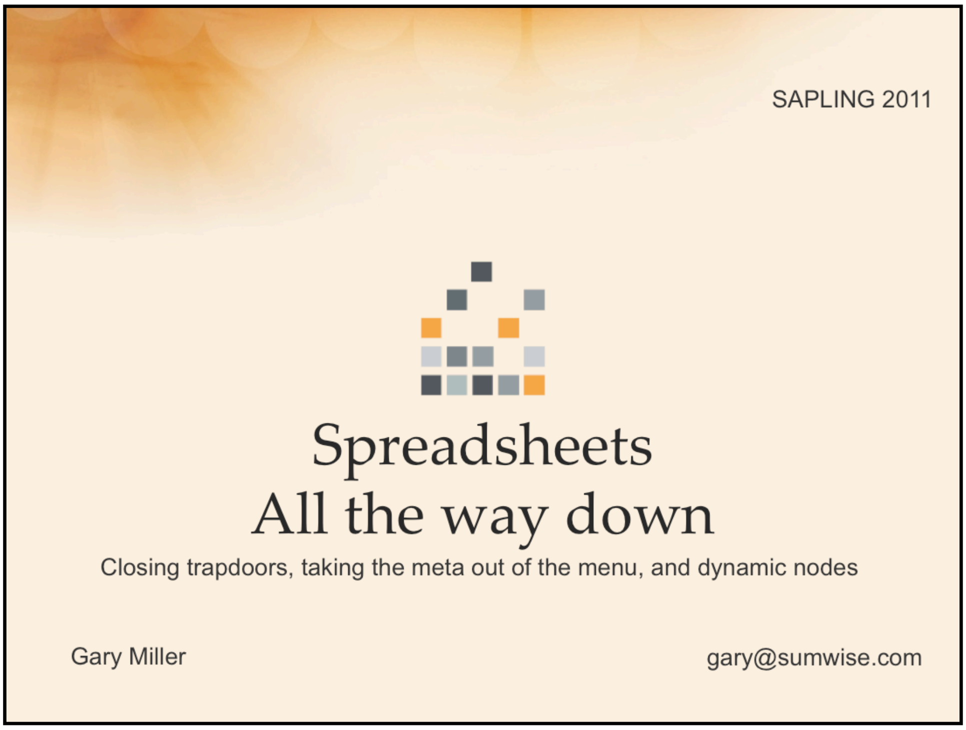


SAPLING 2011

Spreadsheets All the way down

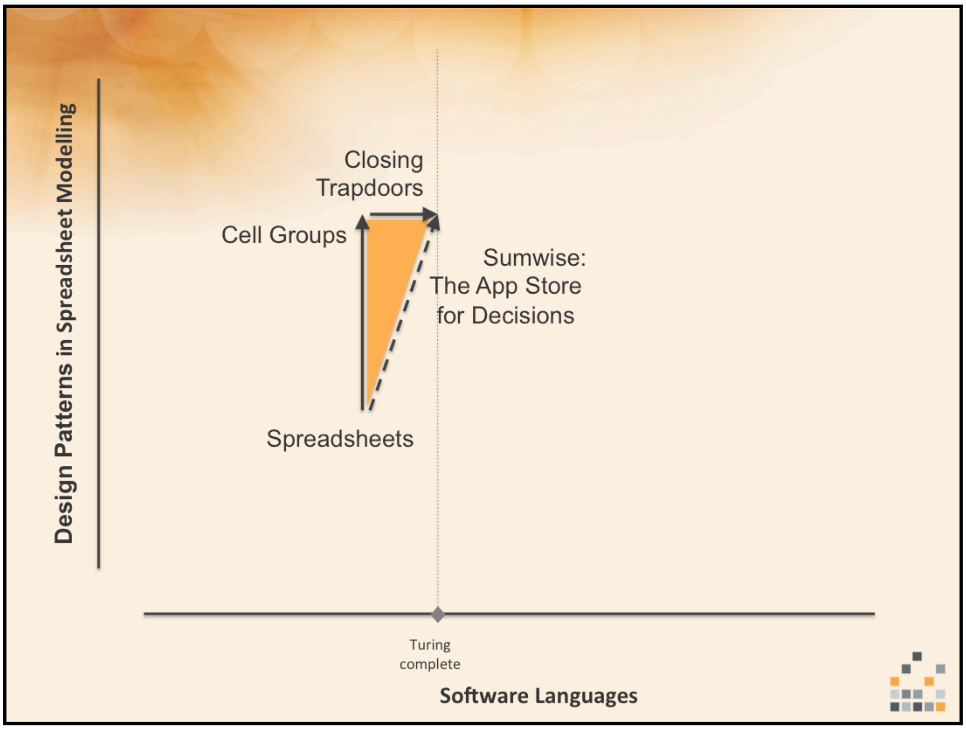
Closing trapdoors, taking the meta out of the menu, and dynamic nodes

Gary Miller

gary@sumwise.com

This talk is mainly focused on the research efforts at Sumwise to create a stronger foundation for decision modelling as typified by Spreadsheets.

Keywords; Programming Languages, Turing Completeness, Spreadsheet


Design Patterns in Spreadsheet Modelling
Closing Trapdoors
Cell Groups
Sumwise: The App Store for Decisions
Spreadsheets
Turing complete
Software Languages

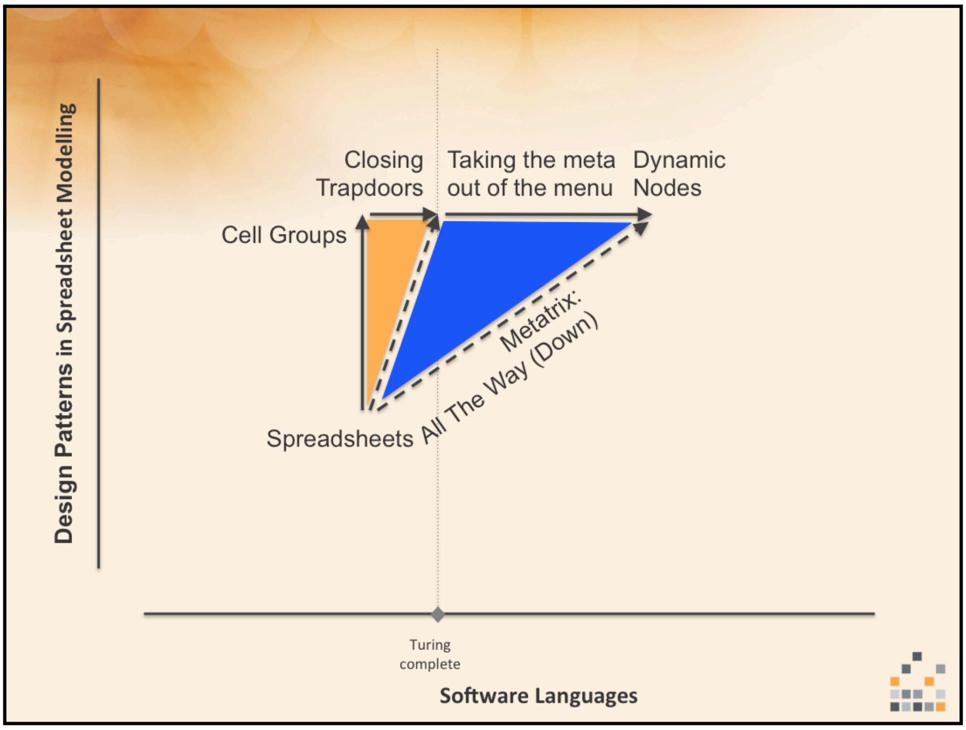
Sumwise’s value proposition is the enabling, with it’s technology platform, of market places for decision models. Content experts with modeling skills can protect and monetize intellectual property created in the form of models. Decision maker can find models or modelers that solve problems and not have to fall back to the default of building bespoke, error prone and substandard spreadsheets.

The development of Sumwise is driven by two forces

1. Capturing the design patterns of expert modelers and making these first class features.
2. Treating spreadsheets as a programming language and learning from existing languages and language evolution techniques.

The features the typify these two forces, currently in the platform, are Cell Groups and Models as Functions. Numerous enabling features had to be added, including; named nodes, structured axes, node tag based meta data, matrix cell values, tuples, internal iterator notation.

Sumwise properties: Testablity, Extensibilty, Comprehensiblity. Turns


Design Patterns in Spreadsheet Modelling
Closing Trapdoors
Taking the meta out of the menu
Dynamic Nodes
Cell Groups
Metatrix: All The Way (Down)
Spreadsheets
Turing complete
Software Languages

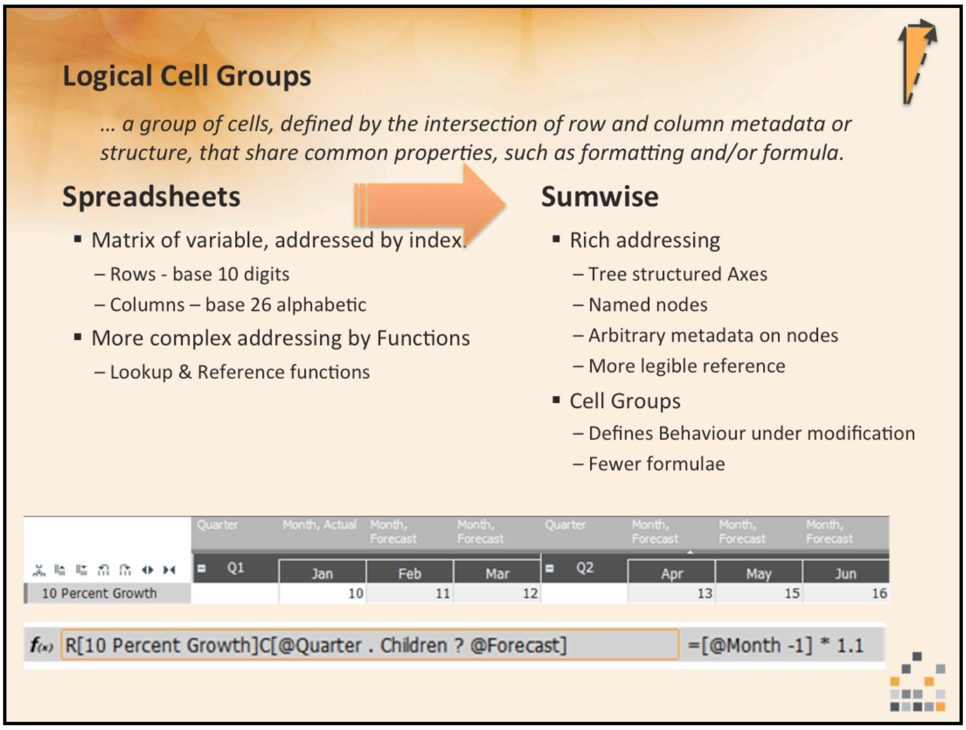
Metatrix is the internal working name of the research being done into underpinning the spreadsheet modeling paradigm, with firm programming language features.

The functions capability in Sumwise is not recursive and therefore leaves spreadsheets in their current state of not being Turing complete (ignoring macro language – "the trapdoor", and circular references - an inadequate and forbidden solution is most cases).

Two orthogonal concepts that have the ability to add computational power past the Turing completeness boundary, and are attractive from a "Conceptual Dimension of Notation", are presented.

1. Passing cell reference by address that can be used by optimization formulae – "Taking the meta out of the menu", and
2. Allowing nodes and tags to not only be named, but upgrading them to formulae – "Dynamic Nodes"

Metatrix features include: Trees as the base building blocks, tree primitives (intersect, attach, filter), optional schematic types, higher order models, meta-circularity, among others.



Logical Cell Groups

… a group of cells, defined by the intersection of row and column metadata or structure, that share common properties, such as formatting and/or formula.

Spreadsheets

- Matrix of variable, addressed by index
  - Rows - base 10 digits
  - Columns – base 26 alphabetic
- More complex addressing by Functions
  - Lookup & Reference functions

Sumwise

- Rich addressing
  - Tree structured Axes
  - Named nodes
  - Arbitrary metadata on nodes
  - More legible reference
- Cell Groups
  - Defines Behaviour under modification
  - Fewer formulae

	Quarter	Month, Actual	Month, Forecast	Month, Forecast	Quarter	Month, Forecast	Month, Forecast	Month, Forecast
	⊟ Q1	Jan	Feb	Mar	⊟ Q2	Apr	May	Jun
10 Percent Growth		10	11	12		13	15	16

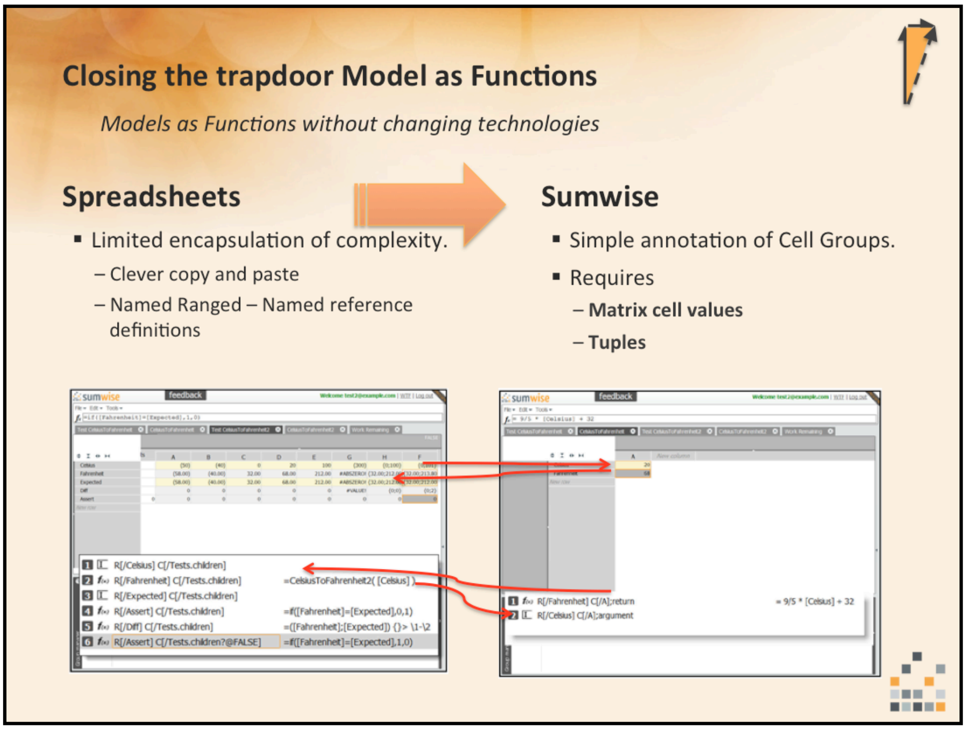
f(x) R[10 Percent Growth]C[@Quarter . Children ? @Forecast] =[@Month -1] * 1.1

What is a spreadsheet

- Matrix of variables, addressed by index
- List of named references – "Named Ranges"
- Complex references achieved via functions

Sumwise Cell Groups

- Allow cells to declaratively receive a formula
- Allows for richer referencing syntax and semantics



Closing the trapdoor Model as Functions

Models as Functions without changing technologies

Spreadsheets

- Limited encapsulation of complexity.
  - – Clever copy and paste
  - – Named Ranged – Named reference definitions

Sumwise

- Simple annotation of Cell Groups.
- Requires
  - – **Matrix cell values**
  - – **Tuples**

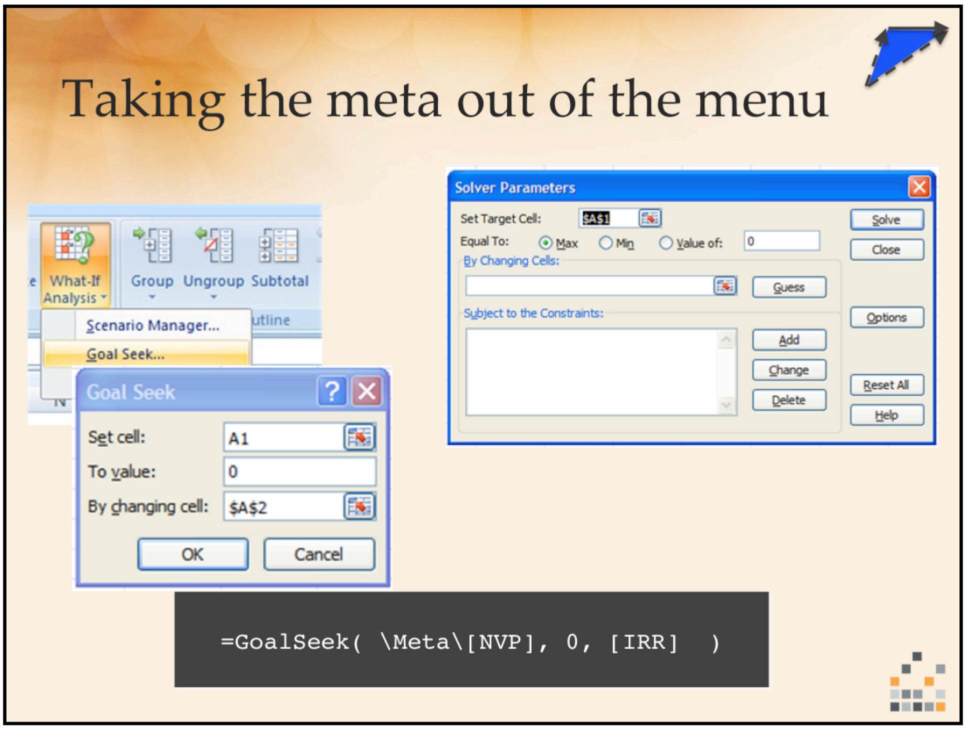
		A	B	C	D	E	G	H	F
Celsius		(50)	(40)	0	20	100	(300)	(0;100)	(0;100)
Fahrenheit		(58.00)	(40.00)	32.00	68.00	212.00	#ABSZERO!	(32.00;212.00)	(32.00;213.80)
Expected		(58.00)	(40.00)	32.00	68.00	212.00	#ABSZERO!	(32.00;212.00)	(32.00;212.00)
Diff		0	0	0	0	0	#VALUE!	(0;0)	(0;2)
Assert	0	0	0	0	0	0	0	0	0

```
1  R[/Celsius] C[/Tests.children]
2  R[/Fahrenheit] C[/Tests.children]             =CelsiusToFahrenheit2( [Celsius] )
3  R[/Expected] C[/Tests.children]
4  R[/Assert] C[/Tests.children]                 =if([Fahrenheit]=[Expected],0,1)
5  R[/Diff] C[/Tests.children]                   =([Fahrenheit];[Expected]) {}> \1-\2
6  R[/Assert] C[/Tests.children?@FALSE]          =if([Fahrenheit]=[Expected],1,0)
```

```
1  R[/Fahrenheit] C[/A];return                   = 9/5 * [Celsius] + 32
2  R[/Celsius] C[/A];argument
```

Models as functions

- Achieved by the annotation of cell group as arguments or returns.
- This implementation favoured from a CDN view point



Taking the meta out of the menu

Goal Seek

Set cell: A1

To value: 0

By changing cell: A2

Solver Parameters

Set Target Cell: A1

Equal To: ● Max ○ Min ○ Value of: 0

By Changing Cells:

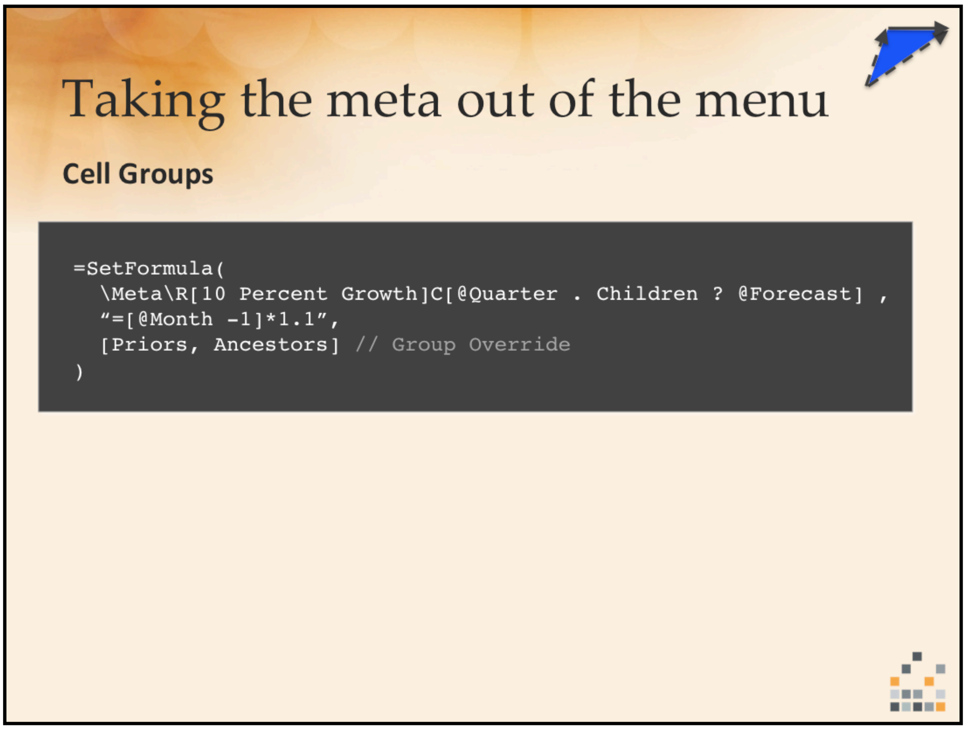
Subject to the Constraints:

```
=GoalSeek( \Meta\[NVP], 0, [IRR]  )
```

What spreadsheet functions can’t be written as Models as functions?

Two classes of these;

1. Part of the machinery. Mainly reference functions, eg. Indirect, address
   - Can be solved with Meta Circularity, topic for another time.
2. Require loops, conditionals (i.e. Turing completeness).
   - Could be solved with recursion, but isn’t very friendly to end-user programmers. Not a good notation for the types of problems being solved, eg IRR.
   - Circular references, but can be non-deterministic – if solution doesn’t converge. Not a neat solution - frowned upon in modelling. Only solves a small set of problems.
   - Metatrix’s preferred solution is References by Address (“Meta” references)
     - Works for optimisation formulae, Currently need to use goal seek from the menu.
     - GoalSeek formulae using “meta” referencing could be placed in a cell or in a one dimensional axis (one dimensional axes are similar



Taking the meta out of the menu

Cell Groups

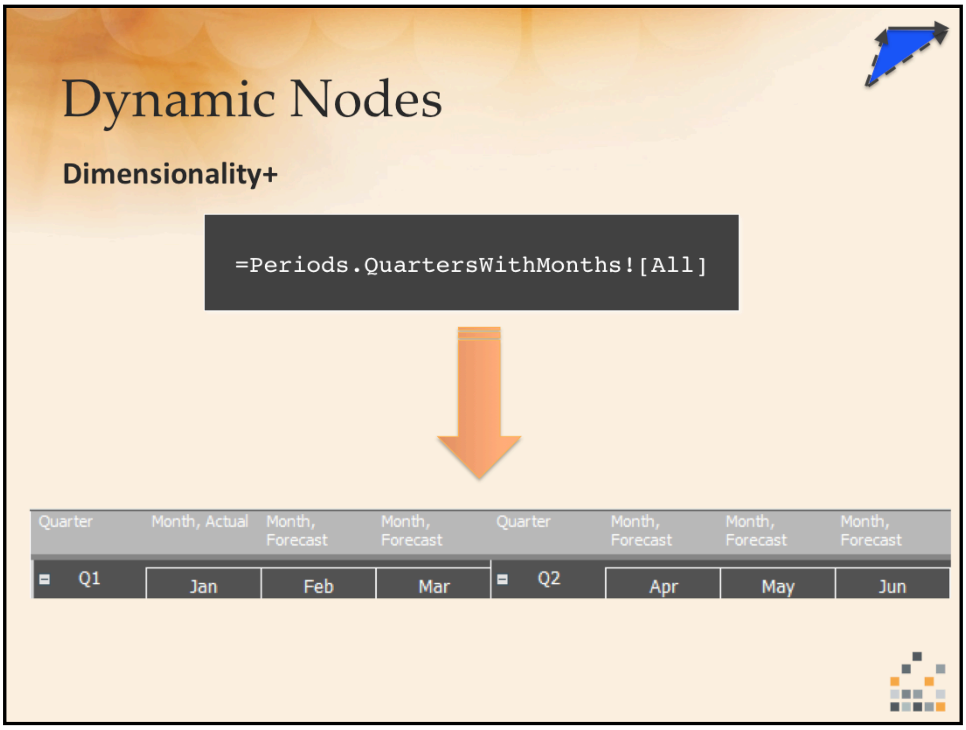
```
=SetFormula(
  \Meta\R[10 Percent Growth]C[@Quarter . Children ? @Forecast] ,
  “=[@Month -1]*1.1”,
  [Priors, Ancestors] // Group Override
)
```

Formula Cell Groups effectively use “meta” references to define the formulae in cells.

- =FormulaGroup(\Meta\R[10 Percent Growth]C[@Quarter . Children ? @Forecast] , “=[@Month -1]*1.1”)

Other Meta that could be taken out of the menu are;

- Creation of nodes. This could be achieved with schemas structure imposed on Axes or Node
  - Nodes create for Schema compliance
    - Eg. Schema applied to an axis. Any node tagged Header must have specified children with specified tags. @Header{Hardware,Software,Service,*,Total[Summary]}
    - Models can be considered as types, Models with schema are statically typed structures.



Dynamic Nodes

Dimensionality+

```
=Periods.QuartersWithMonths![All]
```

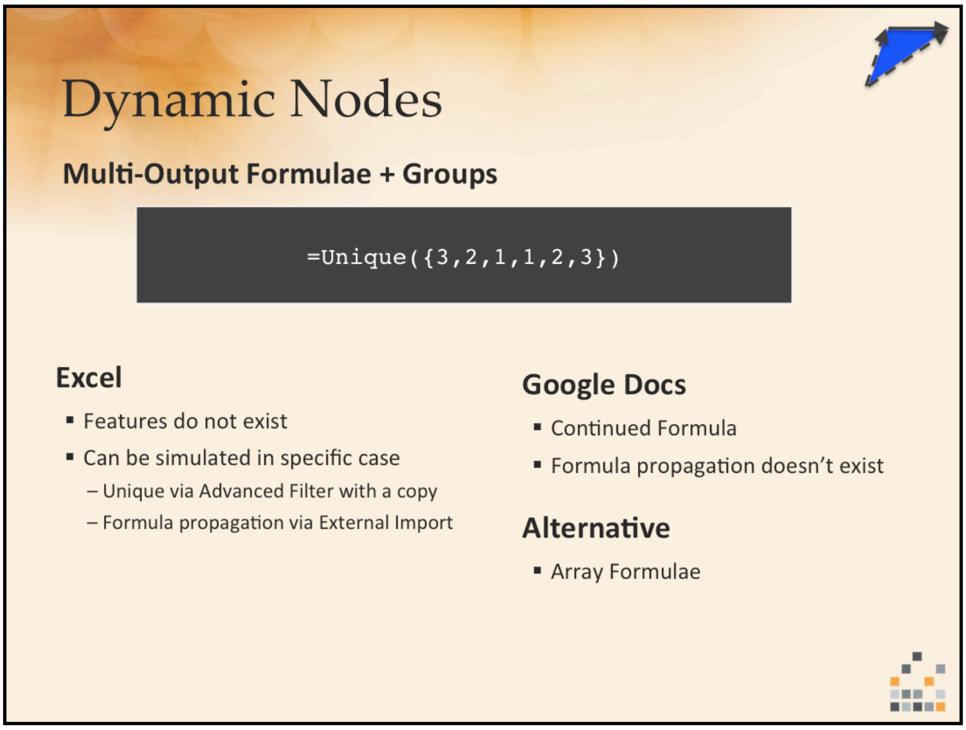
Quarter	Month, Actual	Month, Forecast	Month, Forecast	Quarter	Month, Forecast	Month, Forecast	Month, Forecast
⊟ Q1	Jan	Feb	Mar	⊟ Q2	Apr	May	Jun

Dimensionality:

- Examples of: Lotus Improv, Quantrix, Pivot Tables

Dynamic Nodes go further then this as structure could be referenced.

- These proposed extension are therefore a super of on both spreadsheet and multi-dimensional modelling paradigm.
- A nice enabling feature is linked values, where a change in any number of place is reflected in all.



Dynamic Nodes

Multi-Output Formulae + Groups

```
=Unique({3,2,1,1,2,3})
```

Excel

- Features do not exist
- Can be simulated in specific case
  - Unique via Advanced Filter with a copy
  - Formula propagation via External Import

Google Docs

- Continued Formula
- Formula propagation doesn't exist

Alternative

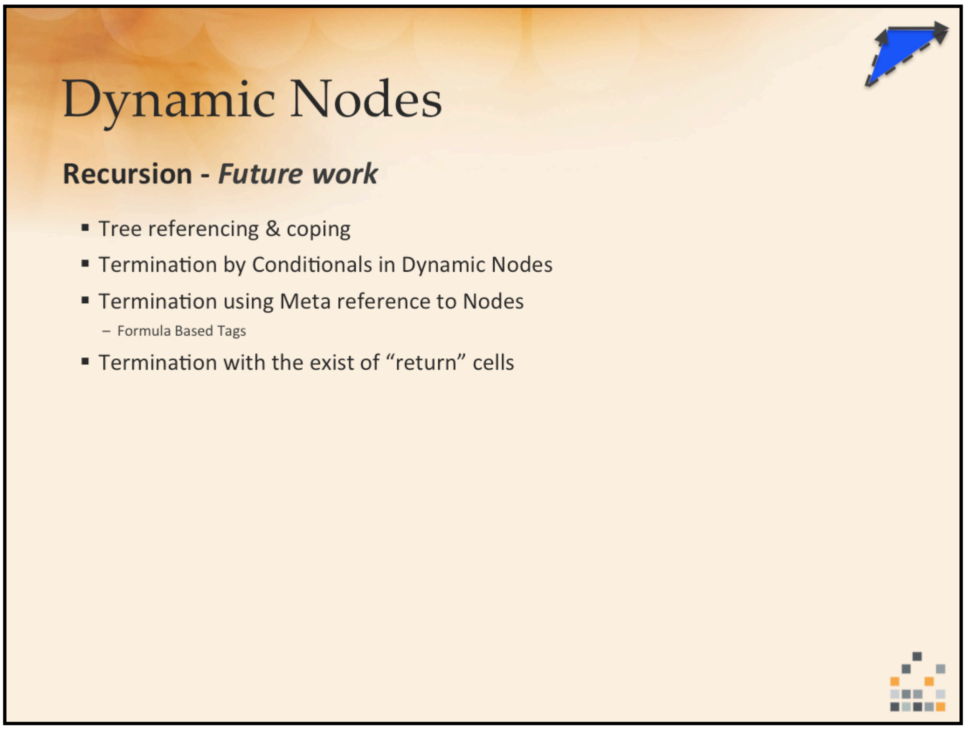
- Array Formulae

Multi output Array Formulae, Filtering, Sorting, Unique list

Repeated Dimension (with structure), Pivot tables, Filtered Views

Combined with the orthogonal notion of Cell Groups leads to highly expressive models.

- Knows how to grow and adapt to modifications
- To some extend can be achieved via Array Formulae, but Dynamic Nodes are score better or CDN
  - Don't need to teach the a new concept (iterators, array formulae …)
  - More consistent



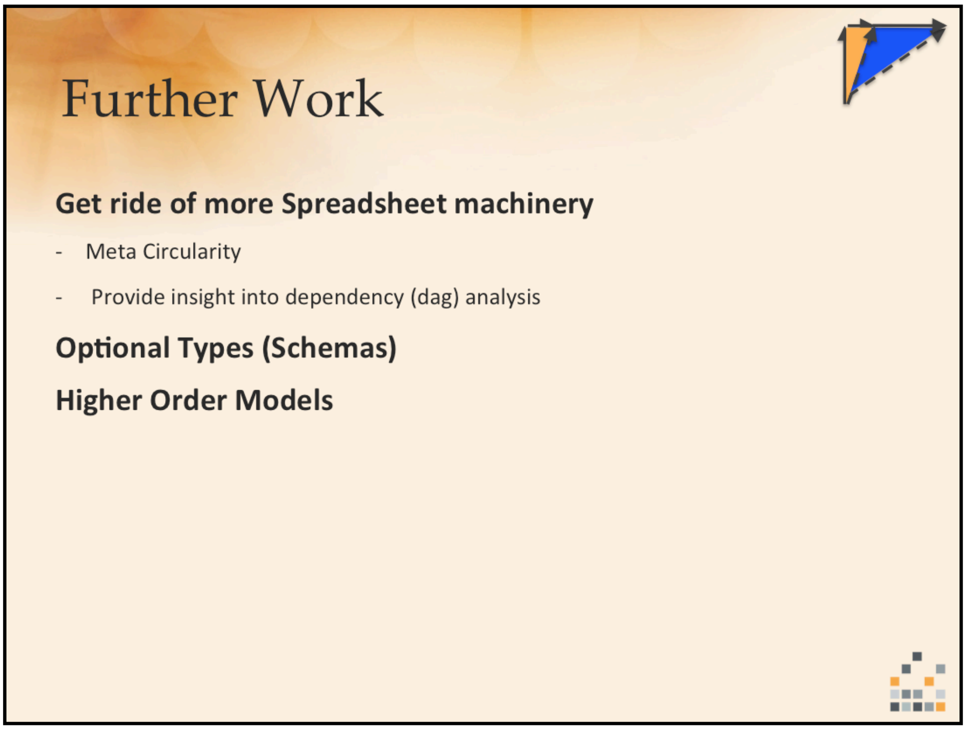
Dynamic Nodes

Recursion - *Future work*

- Tree referencing & coping
- Termination by Conditionals in Dynamic Nodes
- Termination using Meta reference to Nodes
  - Formula Based Tags
- Termination with the exist of “return” cells

Future investigation

- Recursive dynamic nodes
- Node values/formulae defined by “meta” definition in the same way as cell groups
- Dynamic or formula based tags on nodes
  - Used from definition of computation
- These are considered advanced features and might only be used by library writers.



Further Work

Get ride of more Spreadsheet machinery

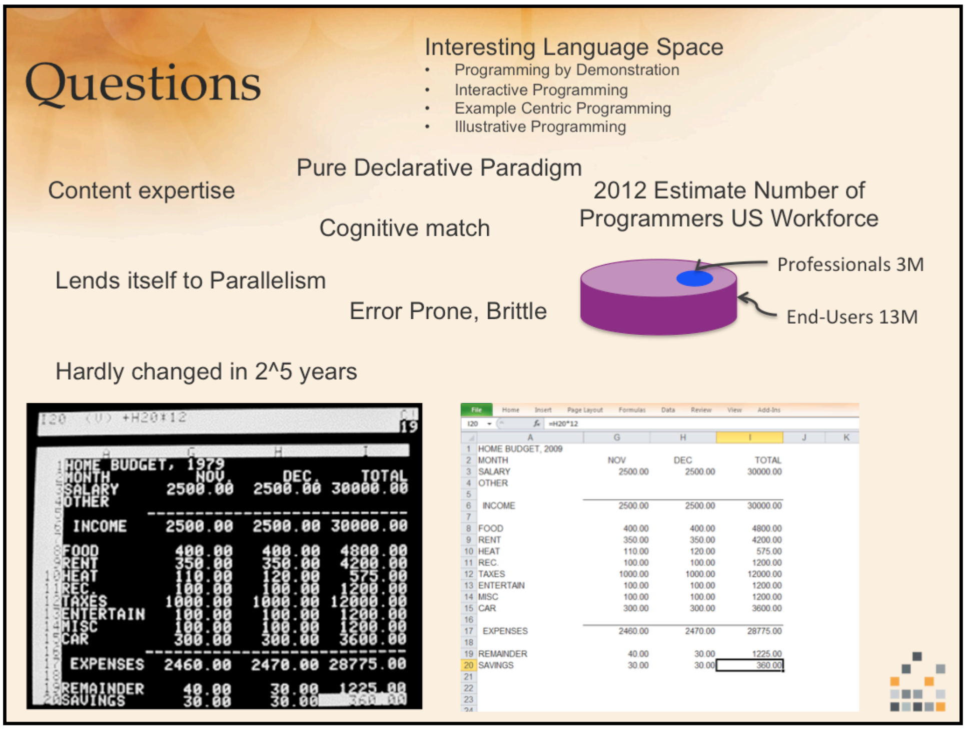
- Meta Circularity
- Provide insight into dependency (dag) analysis

Optional Types (Schemas)

Higher Order Models

Future investigation

- Recursive dynamic nodes
- Node values/formulae defined by “meta” definition in the same way as cell groups
- Dynamic or formula based tags on nodes
  - Used from definition of computation



Questions

Interesting Language Space

- Programming by Demonstration
- Interactive Programming
- Example Centric Programming
- Illustrative Programming

Pure Declarative Paradigm

Content expertise

Cognitive match

2012 Estimate Number of Programmers US Workforce

Lends itself to Parallelism

Error Prone, Brittle

Hardly changed in 2^5 years

```
I20  (V) +H20*12
```

	A	G	H	I
1	HOME BUDGET, 1979			
2	MONTH	NOV.	DEC.	TOTAL
3	SALARY	2500.00	2500.00	30000.00
4	OTHER			
5		----------	----------	----------
6	INCOME	2500.00	2500.00	30000.00
7				
8	FOOD	400.00	400.00	4800.00
9	RENT	350.00	350.00	4200.00
10	HEAT	110.00	120.00	575.00
11	REC.	100.00	100.00	1200.00
12	TAXES	1000.00	1000.00	12000.00
13	ENTERTAIN	100.00	100.00	1200.00
14	MISC	100.00	100.00	1200.00
15	CAR	300.00	300.00	3600.00
16		----------	----------	----------
17	EXPENSES	2460.00	2470.00	28775.00
18				
19	REMAINDER	40.00	30.00	1225.00
20	SAVINGS	30.00	30.00	360.00

```
I20    =H20*12
```

	A	G	H	I	J	K
1	HOME BUDGET, 2009					
2	MONTH	NOV	DEC	TOTAL		
3	SALARY	2500.00	2500.00	30000.00		
4	OTHER					
5						
6	INCOME	2500.00	2500.00	30000.00		
7						
8	FOOD	400.00	400.00	4800.00		
9	RENT	350.00	350.00	4200.00		
10	HEAT	110.00	120.00	575.00		
11	REC.	100.00	100.00	1200.00		
12	TAXES	1000.00	1000.00	12000.00		
13	ENTERTAIN	100.00	100.00	1200.00		
14	MISC	100.00	100.00	1200.00		
15	CAR	300.00	300.00	3600.00		
16						
17	EXPENSES	2460.00	2470.00	28775.00		
18						
19	REMAINDER	40.00	30.00	1225.00		
20	SAVINGS	30.00	30.00	360.00		
21						
22						
23						

That’s it. Questions.

Some reasons I find this an interesting area.