A Pragmatic Approach to Spreadsheet Training

The “Projection-Screen”-Model

Karin Hodnigg

A simple spreadsheet [Panko]

	A	B	C	D	E	F
1	Cash Budget	Assume:		2%	inflation/semester	
2				Fall	Spring	Summer
3	Cash, beginning			$1.000	$1.360	$673
4	Outflows -	School costs		$4.468	$4.474	$4.480
5		Living costs		$4.172	$4.213	$4.485
6	Inflows-	Loans		3000	3000	3000
7		Support from home		$6.000	$5.000	$6.000
8	Cash,end			$1.360	$673	$980
9				=======	=======	=======
10	School (contractual) -	Tuition		$4.115	$4.115	$4.115
11		Fees		$53	$53	$53
12				$300	306	312
13	Living (contractual)	Monthly	Months			
14	Housing	$450	4	$1.800	$1.800	$1.800
15	Insurance	$53	4	$212	$212	$212
16	Living Costs (other)					
17	Food	$330	4	$1.320	1346	1373
18	Entertainment	$150	4	$600	612	624
19	Transportation	$40	4	$160	$161	164
20	Clothing	$21	4	$80	82	84

Spreadsheet Characteristics

- Business Spreadsheets
  - High business value
  - very expensive errors
- "Just Another Office Product" …
- Training based upon
  - Input Values
  - Use functions (e.g. layout, sum-function)
  - Input Formulas
- High learning rates
- Spreadsheets are simple …

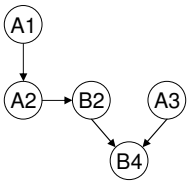
A simple spreadsheet – revisited

	A	B	C	D	E	F
1	Cash Budget	Assume:		2%	inflation/semester	
2				Fall	Spring	Summer
3	Cash, beginning			$1.000	=D8	=E8
4	Outflows -	School costs		=sum(D10:D12)	=sum(E10:E12)	=sum(F10:F12)
5		Living costs		=sum(D14:D20)	=sum(E14:E20)	=sum(F12:F19)
6	Inflows-	Loans		3000	3000	3000
7		Support from home		=round(1000+D4+D5-D3-D6;-3	=round(1000+E4+E5-E3-E6;-3	=round(1000+F4+F5-F3-F6;-3
8	Cash,end			=D3-D4-D5+D6+D7	=E3-E4-E5+E6+E7	=F3-F4-E5+F6+F7
9				=======	=======	=======
10	School (contractual) -	Tuition		$4.115	$4.115	$4.115
11		Fees		$53	$53	$53
12				$300	=round(D12*(1+$D$1);0)	=round(E12*(1+$D$1);0)
13	Living (contractual)	Monthly	Months			
14	Housing	$450	4	=C14*B14	=D14	=D14
15	Insurance	$53	4	=C15*B15	=D15	=D15
16	Living Costs (other)					
17	Food	=11*30	4	=C17*B17	=round(D17*(1+$D$1);0)	=round(E17*(1+$D$1);0)
18	Entertainment	$150	4	=C18*B18	=round(D18*(1+$D$1);0)	=round(E18*(1+$D$1);0)
19	Transportation	$40	4	=C19*B19	=round(D19+(1+$D$1);0)	=round(E19*(1+$D$1);0)
20	Clothing	$21	4	=C20*20	=round(D20*(1+$D$1);0)	=round(E20*(1+$D$1);0)

What are Spreadsheets then?

- "Connected Calculators"?

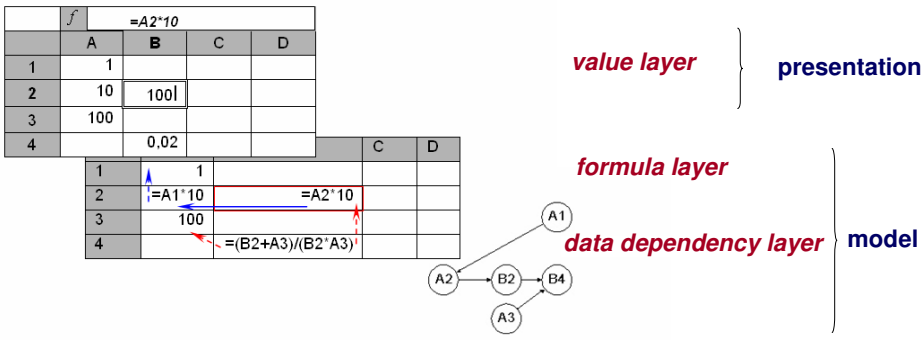
	A	B	C	D
1	1			
2	=A1*10	=A2*10		
3	100			
4		=(B2+A3)/(B2*A3)		



- Programs!
  - "by accident"
  - with a „trial-and-error"-approach
    - „visual check": expected value =?= displayed value

The spreadsheet paradigm

3-level-design of spreadsheets



It is simple
but building a spreadsheet
implies programming!

Constant Change

- Ongoing development
  - Model ruptures (intended/error)
  - Understand the model (Program Comprehension)

	A	B	C	D
1	1			
2	=A1*10	=A2*10		
3	100			
4		=(B2+A3)/(B2*A3)		

?

Document it!

Characteristics

- Persistence of intermediate results
  - No specific, dedicated, unique result
- Difficult loop concept
- Immediate feedback
  - Hides complexity
  - Interactive evaluation process
- Layout is part of the program

Be aware of what spreadsheets are (and what they are not)

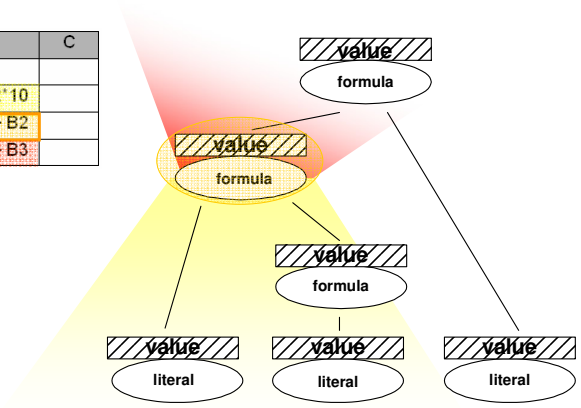
Visibility and Scope

- **Scope of a cell**
  - [complete sheet]
  - Portion of sheet (possibly) affected by change
- **Visibility of a cell**
  - (Transitively) dependent cells
  - Implicit visibility of cells
    - Source cell does not “see” the target cell.
    - A cell “sees” all referenced cells.

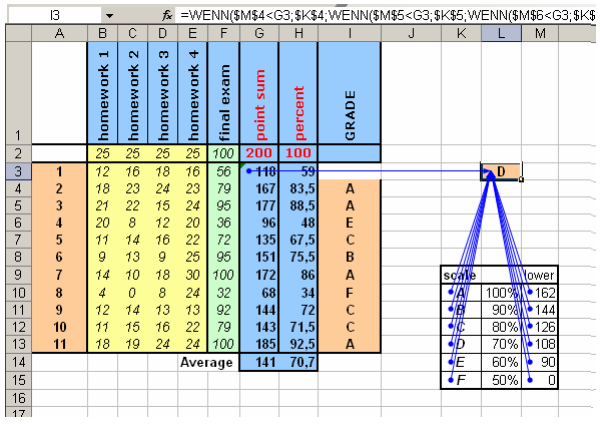
Network of observer cells / observed cells

The Projection Screen Model

	A	B	C
1	1		
2	2	=A2*10	
3	3	=A3 + B2	
4		=A1 + B3	



Patterns: Many-handed figures



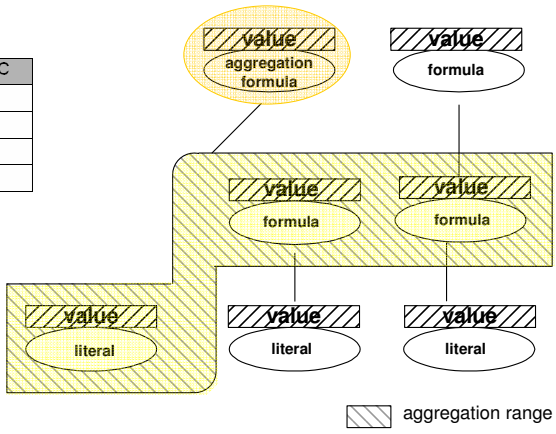
One step after another – solution cell by cell.

Aggregations

	A	B	C	D	E	F	G	H	I	J	K	L	M
1		homework 1	homework 2	homework 3	homework 4	final exam	point sum	percent	GRADE				
2		25	25	25	25	100	200	100					
3	1	12	16	18	16	56	118	59	D		scale		lower
4	2	18	23	24	23	79	167	83,5	A		A	100%	162
5	3	21	22	15	24	95	177	88,5	A		B	90%	144
6	4	11	14	16	22	72	135	67,5	C		C	80%	126
7	5	9	13	9	25	95	151	75,5	B		D	70%	108
8	6	14	10	18	30	100	172	86	A		E	60%	90
9	7	4	0	8	24	32	68	34	F		F	50%	0
10	8												
11	9	12	14	13	13	92	144	72	C				
12	10	11	15	16	22	79	143	71,5	C				
13	11	18	19	24	24	100	185	92,5	A				
14					Average		146	73					
15													
16													
17													
18													

The Projection Screen Model with Aggregation

	A	B	C
1		124	
2	33	=A2*10	
3	12	=A3*10	
4		=SUMME(B1:B3)	



Think, before you do it!

Addressing the Needs of Learners

- Understanding is the goal
  - Modelling, Coaching, Critiquing
- Motivation is the basis
  - Low overhead, immediate success
- Diversity is the norm
  - Collection of various techniques
- Growth is the challenge
  - Adaptable

[Learner-Centered Design, HCI Interactions, 1994]

A Pragmatic Approach

Though simple, building spreadsheets implies programming!

Be aware of what spreadsheets are (and are not)!

One step after another – solution cell by cell!

Document it!

Thus, think before you do it!

Thank you for your attention!

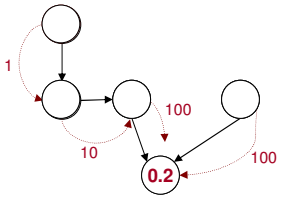
"The limits of my language means the limits of my world."

– Ludwig Wittgenstein –

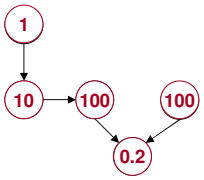
Backup Slides

Semantics

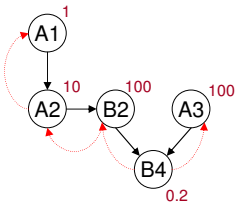
Data Flow Semantics



Graph Reduction Semantics



Spreadsheet Semantics



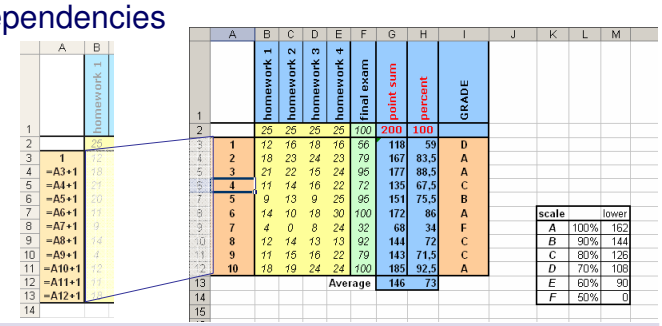
	A	B	C	D
1	1			
2	=A1*10	=A2*10		
3	100			
4		=(B2+A3)/(B2*A3)		

Spreadsheet properties

- tabular layout
- cells
  - containing formula or static value
- declarative formula language
  `=<ref>[<op><ref>]` or `=<func>([<ref>];[param])`
- references
  - single references
  - range references
- immediate evaluation
  - Evaluating references
  - interactive process: change in a cell => re-evaluation

A Spreadsheet language

- Four aspects => programming language
  - + formula language
  - + editor language (layout operations)
  - + data dependencies
  - + layout

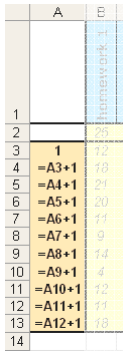


	A	B
1		homework 1
2		25
3	1	12
4	=A3+1	18
5	=A4+1	21
6	=A5+1	20
7	=A6+1	11
8	=A7+1	9
9	=A8+1	14
10	=A9+1	4
11	=A10+1	12
12	=A11+1	11
13	=A12+1	18
14		

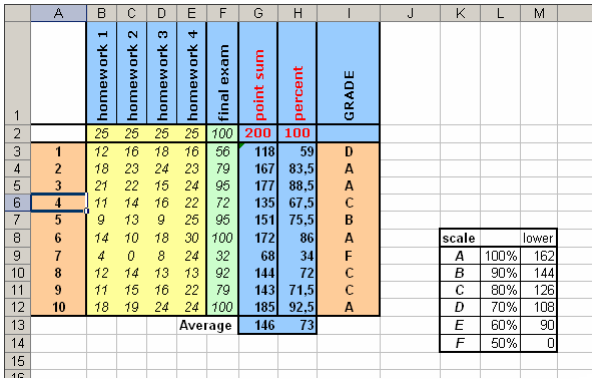
	A	B	C	D	E	F	G	H	I
1		homework 1	homework 2	homework 3	homework 4	final exam	point sum	percent	GRADE
2		25	25	25	25	100	200	100	
3	1	12	16	18	16	56	118	59	D
4	2	18	23	24	23	79	167	83,5	A
5	3	21	22	15	24	95	177	88,5	A
6	4	11	14	16	22	72	135	67,5	C
7	5	9	13	9	25	95	151	75,5	B
8	6	14	10	18	30	100	172	86	A
9	7	4	0	8	24	32	68	34	F
10	8	12	14	13	13	92	144	72	C
11	9	11	15	16	22	79	143	71,5	C
12	10	18	19	24	24	100	185	92,5	A
13					Average		146	73	
14									
15									

scale		lower
A	100%	162
B	90%	144
C	80%	126
D	70%	108
E	60%	90
F	50%	0

Patterns: Queue on a staircase



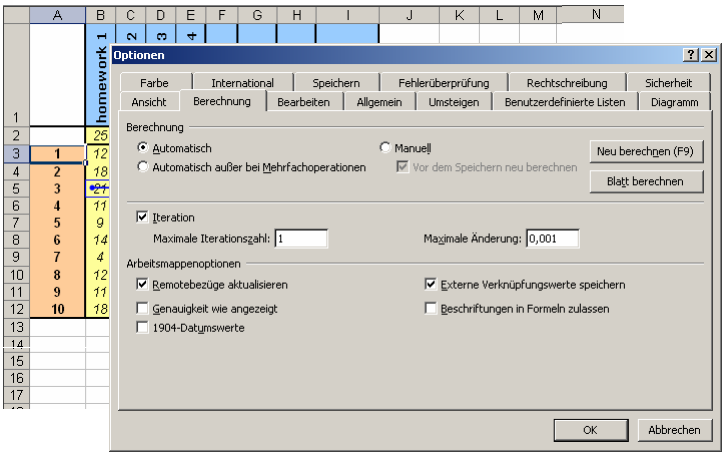
	A	B
1		homework 1
2		25
3	1	12
4	=A3+1	18
5	=A4+1	21
6	=A5+1	20
7	=A6+1	11
8	=A7+1	9
9	=A8+1	14
10	=A9+1	4
11	=A10+1	12
12	=A11+1	11
13	=A12+1	18
14		



	A	B	C	D	E	F	G	H	I
1		homework 1	homework 2	homework 3	homework 4	final exam	point sum	percent	GRADE
2		25	25	25	25	100	200	100	
3	1	12	16	18	16	56	118	59	D
4	2	18	23	24	23	79	167	83,5	A
5	3	21	22	15	24	95	177	88,5	A
6	4	11	14	16	22	72	135	67,5	C
7	5	9	13	9	25	95	151	75,5	B
8	6	14	10	18	30	100	172	86	A
9	7	4	0	8	24	32	68	34	F
10	8	12	14	13	13	92	144	72	C
11	9	11	15	16	22	79	143	71,5	C
12	10	18	19	24	24	100	185	92,5	A
13					Average		146	73	
14									
15									

scale		lower
A	100%	162
B	90%	144
C	80%	126
D	70%	108
E	60%	90
F	50%	0

Patterns: Recursive Images



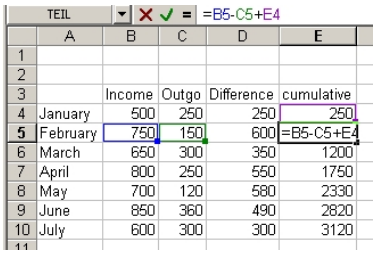
References

- Markus Clermont (2002) “A Scalable Approach to Spreadsheet Visualization”, PhD-Dissertation (University of Klagenfurt).
- Margaret Burnett (2001), “Software Engineering for Visual Programming Languages”, Handbook of Software Engineering and Knowledge Engineering Vol. 2, pages 77-92.
- Takeo Igarashi, Jock Mackinlay, Bay-Wei Chang and Polle Zellweger (1998) ”Fluid Visualization of spreadsheet structures“, Proceedings of the 1998 IEEE Symposium on Visual Languages, pages 118-125
- Roland Mittermeir, Markus Clermon, Christian Hanin (2002) ”A spreadsheet Auditing Tool Evaluated in an Industrial Context“, Proceedings of the 3rd Annual Symposium of the EuSpRIG, pages 35-46
- Jan Erik Moström and David A. Carr (1998), “Programming Paradigms and Program Comprehension”, Proceedings of the 10th Annual Workshop of the PPIG’98, pages 117-127
- Raymond R. Panko (1998) “What we know about spreadsheet errors”, Journal of End User Computing, Volume 10 (2), pages 15-21
- Markku Tukiainen (2001), “Comparing Two Spreadsheet Calculation Paradigms: An Empirical Study with Novice Users”, Interacting with Computers (IWC) , 13 (4), p. 427-446
- Ray Butler (2004), “The Subversive Spreadsheet”, Eusprig Conference Proceedings 2004
- Ayalew Yirsaw, “Testen von Tabellenkalkulationen mit Intervallanalyse”, PhD-Dissertation (University of Klagenfurt).
- K. Rajalingham, D. Chadwick, B. Knight, D. Edwards, „Quality Control in Spreadsheets: A Software Engineering-Based Approach to Spreadsheet Development“, Proceedings of the 33rd Hawaii International Conference on System Sciences

Tool Support

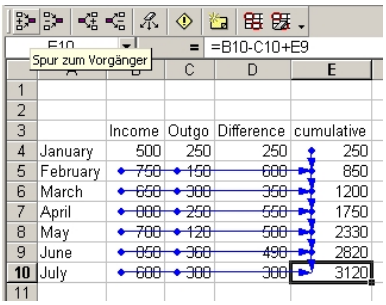
- existing, but inadequate

Editing a formula



	A	B	C	D	E
1					
2					
3		Income	Outgo	Difference	cumulative
4	January	500	250	250	250
5	February	750	150	600	=B5-C5+E4
6	March	650	300	350	1200
7	April	800	250	550	1750
8	May	700	120	580	2330
9	June	850	360	490	2820
10	July	600	300	300	3120
11					

Excel “trace analysis”



	A	B	C	D	E
1					
2					
3		Income	Outgo	Difference	cumulative
4	January	500	250	250	250
5	February	750	150	600	850
6	March	650	300	350	1200
7	April	800	250	550	1750
8	May	700	120	580	2330
9	June	850	360	490	2820
10	July	600	300	300	3120
11					

Spreadsheet properties

- tabular layout
- cells
  - containing formula or static value
- declarative formula language
  `=<ref>[<op><ref>]` or **`=<func>([<ref>];[param])`**
- references
  - single references
  - range references
- immediate evaluation
  - Evaluating references
  - interactive process: change in a cell => re-evaluation

Development Process Specialties

- Automated fill of cells (click and drag)
  - Automated generation of formulas / constants
- Drag-and-Drop
- Copy-and-Paste

Adaptation of parameters

- of formulas
- of references

Conclusion

- Spreadsheet System Implementations are in conflict
  - with established paradigms
  - with each other
- No established conceptual model
  - global vs. local viewpoint

Proper training to raise spreadsheet quality needed!

- "*Project-Screen*"-Model

Relationships and Visibility

- Cell level → functional paradigm
  - Computations in cells are mainly covered.
- Global “picture” → ?
  - proposed model has to meet the basis.
- Visibility and Scope
  - (Transitively) dependant cells
  - Implicit visibility of cells
    - Source cell does not “see” the target cell.
    - Cell “sees” all referenced cells.

Implementing the Spreadsheet Paradigm

Copy/Paste and movement heuristics

- Copy-and-Paste of cells ≠ Moving cells

	A	B	C	D
1	1			
2	2	=A1*10		
3	=A1*10	=A2*10	=C1*10	
4	=A2*10	=SUM(A1:A2)	=C2*10	
5	=SUM(A1:A2)		=SUM(C1:C2)	
6	=SUM(A1:A4)			

range references > distinct cell references > geometrical pattern

Assumptions

- Spreadsheet writers are Non-Computing-Professionals.
- Writing a sheet, they depart from some conceptual model
  - Two categories:
    - Calculators, using features to easily define layout.
    - Layouters, using calculation features.
- The conceptual model is not necessarily stable
  - over persons
  - over time

Consequence

The conceptual model of an individual spreadsheet writer might be implicit,

but if a tool makes something explicit that is close enough to this conceptual model, the writer / an auditor can relate it to the model, easily identify divergences and consider them as symptoms of faults.

Spreadsheet Problems

- *bugs* are everywhere
- *errors in use* are suspected but rarely documented
- *lost opportunities* due to lack of knowledge of Excel tools (e.g., Pivot Table, Goal Seek, Data Validation, …)
- *design processes* are chaotic and inefficient
- *testing* is non-existent
- *standards and training* are minimal or non-existent
- generating *insights* is not well-understood

Spreadsheets = Programs ?

Spreadsheet =
First-order, non-recursive functional program

Correspondence

Is this a problem?

Spreadsheet	Program
Spreadsheet System	Programming Language
Editing (changing formulas)	Programming + Run
Editing (changing values)	Run

Motivation

- Popularität von Tabellenkalkulationen
  - weit verbreitet
  - Datenmanipulation ohne Programmierkenntnisse
    - Fachvokabular
    - Syntax und Semantik der Domäne
- Anwendungen in Tabellenkalkulationen sind
  - intuitiv („Taschenrechner-Prinzip“)
  - leicht zu erstellen
  - plausibel: Sofortiges Feedback *(und Daumen x π)*
- Starke Metapher (Tabelle)
  - reduzierter Lösungsraum durch Vorgabe eines Rahmens
  - gerichtete Problemlösungsstrategien

Spreadsheets als Anwendungen

- Anwendungsklassen
  - Kurzlebige „Ad-Hoc“-Spreadsheets
    - Korrektheit der Werte
  - Langlebige Applikationen
    - Modellentwicklung => Korrektheit des Modells (und der Werte)
    - Wartungsaspekte
- Applikationen mit wirtschaftlicher Bedeutung
  - Pläne, Steuerberechnungen, Bilanzen, etc.
  - meist hoher finanzieller Wert
- Hohe Fehlerraten
  - 25% - 80% Fehlerrate
  - 20-30% nach Test und Debugging
  - ~ 3% aller Zellen enthalten Fehler

Didactical Considerations

- Training must be based on the right paradigms!
- What training methods are in use? (How are SShs taught?)
- Guiding a user (with rigid approaches) vs. freedom
- User’s “mental model” is essential
  - hard to define: domain vocabulary (and metaphors)
  - The conceptual model is not necessarily stable
    - over persons
    - over time
  - Navarro Prieto (1998) found visual images help SS users understood dataflow
  - Petre & Blackwell (2000) note that many programmers report experiencing images