Chapter 2
Requirements

In this chapter, we state the principal requirements for the deductive spreadsheet. We separate them into functional, regarding the operations supported by the tool, and cognitive, concerning the user interactions with the tool. We will rely on these requirements to guide the design of the deductive spreadsheet. They will also be key in a future implementation of a deductive spreadsheet based on this design. The former class will be primarily used in Part I where we design the inferential engine of the tool. The latter will play an important role in Part II, where we extend the interface of a traditional spreadsheet to provide access to the new deductive facilities.

We give the functional requirements of the deductive spreadsheet in Sect. 2.1, and present its cognitive requirements in Sect. 2.2. How to evaluate these requirements is discussed in Sect. 2.3.

2.1 Functional Requirements

This class of requirements is concerned with the functionalities that the design makes available. They concentrate on the supported operations and how they interact among each other. They also assess abstract issues such as expressiveness and computational cost. These requirements do not encompass the manner in which functionalities are presented to the user. In particular, design decisions such as concrete syntax and visual feedback are not discussed here, nor is a method for evaluating them.

The principal functional requirements for the deductive spreadsheet are described next.

I. Cervesato, *The Deductive Spreadsheet*, Cognitive Technologies,
DOI 10.1007/978-3-642-37747-1_2, © Springer-Verlag Berlin Heidelberg 2013

F1. Functional Conservativity

Every traditional spreadsheet functionality shall still be available in the deductive spreadsheet. In particular, each of the following notions and the associated classes of operations shall remain supported:

1. Cells, scalar values and scalar formulas.
2. Arrays and array formulas.
3. Relations (known as data lists in Microsoft Excel, for example) and their supporting operations, such as filtering and sorting.
4. Explanation and auditing.

The support for relations will be naturally enhanced thanks to the deductive extension.

F2. Expressiveness

The deductive spreadsheet shall embed symbolic reasoning capabilities at least as expressive as the Datalog fragment of first-order logic programming. In particular, the following notions shall be supported:

1. Logical formulas in the Horn clause fragment of first-order logic with a term language deprived of function symbols. In particular, support will be available for recursive clauses.
2. Stratified negation in the body of a clause.
3. Interpreted comparison predicates, in particular, equality.
4. A fragment of a functional language, appropriately restricted so as not to violate any other functional requirement.

F3. Supported Inferences

The deductive spreadsheet will support yes/no queries, queries returning values, and explanation. Specifically:

1. The user shall be able to ask for all the values that satisfy a given logical formula. This encompasses yes/no queries as a special case.
2. Logical inference will return all solutions to these queries without duplicates.
3. The user shall be able to ask why a certain value was returned and the system will trace the inference back to established facts, both by unfolding the reasoning one step at a time, and by showing all inferences at once.
4. The user shall similarly be able to ask why a certain value was not returned.

F4. Termination

Logical inference shall produce all results in a finite time. In particular:

1. The logical query language of the deductive spreadsheet shall be decidable. Datalog satisfies this property. The proposed extensions of Datalog shall retain it.
2. The decision procedure implementing the reasoning capabilities shall not allow infinite loops. Bottom-up evaluation achieves this for Datalog, and will be retained in the proposed extensions.
3. The decision procedure shall report all results to a query without duplicates. Again this is a characteristic of bottom-up evaluation in Datalog that shall be preserved.
4. The computational expense of processing a query shall be comparable to or not overwhelmingly larger than the cost of evaluating a traditional spreadsheet.

F5. Updates

Changes in cell contents (including formulas) or inference rules shall be immediately reflected in any deduced value, unless automatic re-evaluation has been disabled by the user. Namely:

1. Modifying a value contained in a cell shall propagate to the functional part of the spreadsheet as well as to any deduced value that relies on it, even indirectly.
2. Modifying a scalar or array formula shall similarly propagate the resulting recalculated values to all parts of the spreadsheet that depend on it, whether in traditional formulas (functional dependency) or in calculated relations (deductive dependency).
3. Modifying a logical formula shall instantly lead to its re-evaluation and changes in the resulting computed or deduced values shall propagate to all parts of the spreadsheet that depend on it, either functionally or deductively.
4. The user shall have the ability to disable automatic re-evaluation if desired.

F6. Functional Integration

The deductive and traditional components shall be integrated, so that values computed deductively can be part of formulas and values calculated from formulas can participate in deductive tasks.

1. A logical formula shall be able to draw its input from cells whose value has been calculated through a scalar or an array formula.

2. A scalar or array formula shall be able to draw its input from cells whose value has been deduced by means of a logical inference.
3. A logical formula may embed a scalar or array formula, subject to the constraint that no other requirement is violated.
4. A scalar or array formula may embed a logical formula, subject to the constraint that no other requirement is violated.

2.2 Cognitive Requirements

This class of requirements has to do with the general experience of the user during his or her daily interactions with a tool. Therefore, issues that are addressed include the manner in which the graphical interface provides access to the underlying functionalities, the familiarity with respect to similar applications, and the effort involved in using the tool and discovering new features.

The principal cognitive requirements of the deductive spreadsheet are listed next.

C1. Cognitive Conservativity

All traditional functionalities shall be accessible to the user in the exact same way as in a traditional spreadsheet. The deductive interface shall therefore be conservative and unobtrusive. In particular:

1. All traditional methodologies for entering data and formulas in a spreadsheet must be supported by the deductive extension. In particular, values and formulas can be entered in a cell as text using a keyboard. Formulas can also be entered using menus and "wizards" in conjunction with cell range selection using the mouse. The intelligent cut-and-paste functionality shall be available as in the traditional spreadsheet to iterate formulas to new regions of the spreadsheet.
2. All traditional methodologies to display calculated values and other feedback shall be supported. This includes the standard tabulated display of calculated cells, and visual rendering using charts and other graphical methods.
3. The user shall not be penalized by the deductive extension whenever he or she is not using it. In particular, no time overhead should be incurred when using only the traditional functionalities of the deductive spreadsheet.

C2. Consistency

The commands supporting the deductive extensions shall be consistent with the existing modalities of interaction and with the user's expectation. Therefore:

1. The user shall be able to enter logical formulas as text using the keyboard.
2. He or she shall also be able to make use of the mouse to select ranges and specific operations.
3. Menu items and "wizards" shall also be available consistently with the traditional functionalities.
4. The intelligent cut-and-paste functionality shall be supported and operate on logical expressions in the same way as for scalar formulas.
5. These commands shall be intuitive and easy to use.

C3. Cognitive Integration

Using the deductive capabilities shall be natural and intuitive, including when used in conjunction with the traditional functionalities. Specifically:

1. The syntax supporting the deductive functionalities shall be chosen so as to have an intuitive meaning even for users unfamiliar with logic.
2. The textual and graphical syntax available to access the deductive functionalities shall be consistent with the input methods for traditional formulas, and similarly for output.
3. There should be as few barriers as possible preventing a user from embedding traditional formulas in a deductive expression and vice versa.

C4. Performance

For common forms of data inference, an implementation of the deductive spreadsheet should have a performance that is subjectively comparable to spreadsheet operations of intermediate complexity. Specifically:

1. Update propagation should feel instantaneous, both for traditional and deductive formulas.
2. Evaluation from scratch should not take longer than loading a midsize traditional spreadsheet.
3. A deductive spreadsheet that does not use deductive formulas should have the same subjective performance as a traditional spreadsheet.

C5. Discovery

The deductive extension should have a gentle learning curve, so that users can reinforce and extend their skills by simply using the tool and minimal experimentation. In particular:

1. Pull-down menus, buttons, right-mouse-button menus and "wizards" should unobtrusively make deductive operations and concepts available to the user.
2. Once exposed to the basic deductive task of entering a simple conjunctive formula, entering more complex formulas, e.g., disjunction or negation, should quickly become natural.
3. The user should easily grasp the possibility of using the output of a logical query as the input of a scalar or array formula, and vice versa.
4. The syntax for embedding traditional and logical formulas within each other should be intuitive.
5. Error messages should be helpful in guiding the user toward proper use, not just away from unavailable options.
6. Supporting material in the form of an integrated help, a user manual and possibly a step-by-step tutorial should be available and easily accessible whenever needed.
7. In the long run, Internet support, both from the implementers and a user community, is expected.

2.3 Evaluation

Functional requirements are assessed by checking that the design of the deductive apparatus meets them. This can be done by inspection for a number of minor requirements especially when they are not interdependent. Some more theoretical requirements, such as decidability, are often established in the literature or by extending arguments found in the literature. Verifying functional requirements that heavily depend on each other generally involves substantial analysis.

Cognitive requirements need to be assessed empirically and experimentally. Methodologies to do so abound in the literature. We chose to base our design on two techniques, the theory of cognitive dimensions of notations [GP96] and the attention investment model [Bla02], both of which had been applied with good results in other extensions of the spreadsheet [BAD+01, PBB03]. These techniques are intended to assist a designer who is not an human-computer interaction expert in the task of creating a plausible preliminary user interface. An actual evaluation of our design needs to rely on rigorous empirical assessment methodologies based on predictive modeling and user testing.

http://www.springer.com/978-3-642-37746-4

The Deductive Spreadsheet
Cervesato, I.
2013, XXI, 406 p. 98 illus., Hardcover
ISBN: 978-3-642-37746-4