Role of Variables in KIM

Expressions, variables and functions are key concepts to KIM. They stand behind any definition of a laboratory test. For majority of “simple” methods, one need not bother about those concepts as a simple, form-like interface is available to generate underlying formulas. The user just fills-in names of categories and their limits. The system then transforms this input into a series of expressions that are understood by KIM. Yet, there are situations, when this simple definition mode is not powerful enough to describe more sophisticated tests. Then the user has to enter his formulas directly. He then have to understand some basic concepts of information technology.

A user of KIM is supposed to have a basic mathematical background while he is not supposed to be familiar with concepts related to information technology / programming (at least current generation of users). Following text tries to explain some concepts that (as we feel) may be difficult for such a user.

Let’s start our discussion with Pythagoras. Everybody has seen the equation of:

C*C = A*A + B*B (Instead of X square we write X*X, which is equivalent)

The sum of squares of sides A and B equals to the square of the third side C of a triangle. Consider the right hand side of the equation. It says: Take a length of side A, square it, take a length of side B, square it and sum the two squares. It is an expression. One understands, that symbols A and B stand for some numbers. To check whether a triangle has a right-angle, one will measure its sides and will substitute the numbers in place of A, B and C symbols.

In KIM, **variables** take the role of mathematical symbols.

A **variable** is not only a symbol in mathematical sense, it is also a place-holder for a value. Above that, a variable has its type – type of value it contains (either a number or a text). It may also have a “structure” – it contains either a single value (a single number or a single text) or it may also contain a list of values (several numbers or several texts). We differentiate a “simple” variable (containing a single value) and a “vector” (containing a list of values).

In mathematics, no one cares about above mentioned characteristics of a symbol. That’s is why a term “symbol” is not sufficient and we have to accept a concept of **variable**.

Categories of Variables in KIM

In KIM, we can differentiate between several kinds of variables. Some are supplied by the system, others are defined by the user.

- Plate Data

When you use constructs such as A1, B1, C3 etc... (a letter A..H followed by a number 1..12), the system understands them as variables containing measured (normalized) data at specific plate wells. After a plate was measured, those variables will get their values (absorbances at specific wells). It is also possible to put those names together (e.g. A1C1, H1H12 – there is NO space between A1 and C1, H1 and H12). Then such variables will contain lists of values (A1C1 contains values in wells A1, B1 and C1). Then there are function that can operate on lists of values.

e.g.: avg(A1C1), avg(A1,B1,C1)

The avg() function calculates an average value of its arguments and those argument may be both single values or lists of values. Thus the above examples will lead to the same result.

- Sample Names

When in the layout you place a control of **Pos**, its name may then be used in an expression. There it is considered as a variable containing absorbace(s) in respective well(s). The value of the variable will be set after data were measured. If the **Pos** control appears just in a single well, then **Pos** will contain a single number. When there were more **Pos** controls, then the variable **Pos** would contain a list of numbers.

- User Variables

User variables are introduced by the user when he defines a test. Whenever you enter an expression in either calculations, evaluations or validations, the system generates variables that will contain results of those expressions. Those variables are under the control of the user. He may choose his own names for those variables. Types of user variables (a number or a text) are derived from their defining expressions.

- Other Variables

There are few more variables that the system supplies and that do not fall into above groups.

e.g.:

gblank ... value used for blanking a plate

conc ... concentration of a “quantitative” sample

Yet more variables are introduced in reports.

Expressions May Contain Texts !

While the user easily accepts that in expressions he specifies calculations with numbers, he may be surprised that also texts may be used in expressions and that it is possible to calculated with them. Consider the following example:

A + B

If A contains a value (number) 1 and B contains a value 2, then the result will be a number 3.

BUT: If a variable A contains a text “Good” and B contains “Bye”, then the operator of ‘+’ has a meaning of “concatenation” of texts and the result will be a text: “GoodBye”.

In some situations (actually in many situations) the description of a result is constructed from a pieces of texts as in:

str(A1) + “ OD”

Remember: A1 is a variable containing an absorbace in well A1. It is a number. The str() function converts that number into a text. Only then we can use the ‘+’ operator to concatenate the two texts. If A1 contains a value of 1.034 then the result of the above expression will be a single text:

“1.034 OD”

--- END OF TEXT ---