The drugulator - a new method for performing dosage calculations

eJHI
electronic Journal of Health Informatics
http://www.ejhi.net
2009; Vol 4(1): e7

The Drugulator – A new method for performing dosage calculations

Ronald C Monson[1]

[1] Language and Learning,Victoria University, Melbourne, Victoria, Australia

Abstract

In this paper we introduce The Drugulator - a new method for performing dosage calculations at the administration stage of a patient's medical management. In this method users first construct a dosage's mathematical formulation from within the Drugulator's structured interface. This formulation is then evaluated by sending it to an attached computational backend that then returns any constituent calculations along with the patient's final dose. This process of Formulaic and Computationally-Aided Dosing (FCAD), and in particular, the single-interface Drugulator implementation described here, makes it practical to readily automate a wide range of dosage calculations at the point of care. This practicality also enables further automation through a progressive integration with existing medical systems. Finally, we argue that the principle underpinning the deployment of a single interface can also be used to improve the usability of those computer provider order entry systems that include clinical decision support (CPOE$_{cds}$).

Keywords: Drug dosage calculations, medication errors, educational technology, algorithms, automation, systems integration

1. Introduction

The main purpose of this paper is to present the Drugulator as a viable clinical tool for calculating drug dosages at the administration stage of the medication process. To this end, its basic operation and underlying principle will be described, illustrated and discussed with several examples. There is also an initial web implementation[1] where these descriptions can be experienced first-hand and a wider range of examples consulted from the accompanying user manual. A more general goal of the paper is to highlight the need for more research into the application of FCADs at this stage of the administration process and how they can be integrated with medical systems that automate other stages. As part of this, several metrics are developed as a way to initially evaluate FCADs (and are applied to the Drugulator), while the Drugulator's underlying *Constructor* principle is developed and used to show how such systems can be both applied to, and connected with, CPOEs.

1.1. Rationale

The most significant and immediate application of FCADs like the Drugulator is as the *calculating component* of a system designed to address the current, unacceptably high levels of Adverse Drug Events (ADEs) in hospital systems. This need follows from the by now well-documented analyses of the casualties and associated costs - we quote again some of the more sobering statistics: In the US, it is estimated that in a large 700 bed hospital, medical errors cost $5.6 million[2] annually while over a decade ago the total cost of drug-related errors in the US alone was put at 76 billion[3]. On the human side, it has been estimated that ADEs make up 19%[4] of the general medical errors that are responsible for between 44000 & 98000 deaths a year - a higher mortality rate than that of car accidents, suicide, homicide or AIDS[5]. Finally, there is evidence that if you enter a hospital as a patient, there is a 6.5-11% chance that any proffered medicine will increase your hospital stay by an additional 2.2 days or a 0.065% chance the accepted medication will contribute to your death[2, 6-9].

The process through which ADEs occur is clearly multi-faceted but one framework designed to pinpoint critical junctures is a medical management model that describes 5 stages - *Prescribe->Transmit->Dispense->Administer->Monitor*[10]. Of these

The *electronic Journal of Health Informatics* is an international journal committed to scholarly excellence and dedicated to the advancement of Health Informatics and information technology in healthcare. ISSN: 1446-4381

stages, there is consistent evidence that the most significant stages in terms of ADEs' root causes are the *Prescribe* and *Administer* stages[10]. For example, the *Prescribe* stage has been variously estimated as being responsible for: 39%[8], 43%[11], 56%[7], 58%[12], 59%[8], 61%[13], 79%[14], percent of ADEs while the *Administer* stage has been variously estimated as contributing to such events: 4%[14], 13%[13], 21%[12], 34%[7], 38%[8], 44%[15], 52%[11], 57%[8] of the time. In this paper, we focus on the *Administer* stage and, in particular, the *calculating* component used to generate a correct dosage within a FCAD.

Failure to reach the correct dosage has been variously identified as contributing to 36%[16], 58%[10], 60%[17], 67%[15] of the errors in the *Administer* stage with the calculation process a central component in this failure. While we focus on the calculating component of the *Administer* stage and hence the way in which the Drugulator can assist nurses in such calculations, the underlying principle used by the Drugulator also has relevance to the *Prescribe stage*. Indeed, it will be argued that applying this principle to simplify the interfaces of CPOE systems is a way to maintain a certain level of usability in the face of increasingly complex prescriptions that incorporate larger numbers of variables.

The main way in which it is envisioned that the Drugulator can reduce ADEs is through health-care workers using its *single* machine interface to perform *all* their dosage calculations in a more efficient, accurate and ultimately safer manner. The words *single* and *all* have been emphasized here since we argue that they represent the Drugulator's key advantage over both traditional pen/paper/calculator methods and the suite of existing dosage calculators. The advantage of this single interface is that it overcomes serious usability issues arising from the use of multi-interface arrangements in current systems.

Here and for the remainder of the paper “Single Interface” should be interpreted as short-hand for *sufficiently usable* Single Interface. This qualification is necessary because a collection of different interfaces can always be combined into a single but now more complex interface. While this results in a single interface (that contains all the possible calculations previously contained in the different interfaces), the trade-off is a loss of usability given that users now have to negotiate within a screen of far greater complexity. This trade-off in usability is made more precise later in the *Efficiency Index* section in which the Drugulator advantage can be seen by the way in which it minimizes the number of persisting fields while at the same time maximizing the number of calculations that can be performed using such fields.

A central component of this usability relates to the efficiency associated with the average time needed to perform a dosage calculation. In CPOE systems (for which more research has been conducted) this time factor has been identified as critical. For example, in one review the first of Ten Commandments to implementing effective CPOEs is *Speed is Everything*[18] and this exhortation has continually informed best practice[19] and the generation of Order Sets[20]. Further, its lack has been identified as a key barrier to the adoption of hand-held medical devices[21]. While this impetus has been researched mostly for the *Prescribe* stage, there is emerging evidence that nurses in the *Administer* stage face similar time pressures[16, 22] that can, to some extent be addressed by the introduction of FCADs.

Another application for the Drugulator is in the education and training of health-care professionals involved in point of care drug administration. This includes those who will not necessarily be accessing the Drugulator as part of their clinical practice but who can nonetheless benefit from using its interface as part of their training in traditional pen/paper/calculator procedures. In particular, by mastering the Drugulator's operation, apart from being able to check answers arrived at in the traditional way, learners are able to gain an insight into the single principle that underpins a wide range of calculations. Finally, the other more long-term application for the Drugulator is that its method, through its interface design, provides opportunities for ongoing reductions in ADE occurrences via a pathway leading to a further automation of point of care drug-delivery. This pathway involves the integration of existing and emerging hospital systems by leveraging a certain extensibility that is inherent to the method and is explained later in the section *A Developmental Blueprint*.

1.2. Comparison with Existing Methods

Calculating a patient's dose at the *Administer* stage can be done in various ways and the following is a rough categorization of the existing methods: 1) Inspection 2) Pen/Paper 3) Pen/Paper/Calculator and 4) Computer/PDA. Method 1 involves a dosage calculation being performed in a user's head and is therefore usually appropriate for straightforward tablet calculations. Method 2 involves users constructing the complete mathematical expression before evaluating it by carrying out the consequent arithmetical manipulations on paper. Method 3 is essentially Method 2 except that a calculator is used to evaluate the constructed expression. Method 4 includes a range of computational tools/dosage calculators that have emerged over the last decade. These range from freely available online calculators[23] to commercially released systems[24] that form part of more generalized health care tools. The actual uptake of these tools in the *Administer* stage however, has not been as pervasive due, as we will argue, to a specific usability issue related to a FCAD's *efficiency*.

Method 3 has been by far and away the most common of all the methods over the past 50 years with various curriculum programs and clinical procedures instituted as befitting such a long-standing practice. Despite these programs however, the remaining high ADE levels, or at least those attributable to drug calculation error, constitute persuasive evidence of these programs' inherent limitations and suggest the need for a new approach. In this paper we take the position that the additional layers of automation and checking possible in Method 4, if feasible in the clinical situation, *will* ultimately represent an improvement (in a similar way $\mathrm{CPOE}_{\mathrm{cds}}$ are beginning to improve the *Prescribe* stage) but acknowledge that further studies will be needed as confirmation. Hence, any comparisons of the Drugulator for the remainder of this paper will be less with the status-quo of Method 3 and more with the interface design of existing dosage systems within Method 4.

As mentioned previously, the fundamental difference between the Drugulator and existing dosage calculators is its use of a single interface. The most important and immediate effect of this single interface approach is in establishing the feasibility of automating point-of-care dosage calculation in a general way. That is, the current approach of searching amongst a large number of interfaces (each corresponding to a possible dosage calculation) is fundamentally limited by the average time such a process takes. The point of the Drugulator is that such a search becomes unnecessary since the required dosage calculations can be conducted from its single interface at the point of care.

There are other less fundamental differences between the Drugulator and other systems (e.g. features that are transferable between the various interfaces) but that can nonetheless affect their usability and therefore their general ability to reduce ADEs. These differences include such features as dosage output form, internal transparency, pharmacopoeia integration, server-side architecture, computational engine, general workflow etc. Some of these features are introduced for the first time in the Drugulator whilst some other features from more established systems could no doubt improve the Drugulator's overall effectiveness. These comparisons are discussed in turn as they are introduced as part of the Drugulator architectural description in the *System Description* section and in the Appendix.

2. System Description

2.1. Status Report

The Drugulator was conceived by the author while working in the Portfolio of Language and Learning at Victoria University (VU). This portfolio targets learning support in problematic areas identified by the University’s faculties and schools. In 2005, The School of Nursing and Midwifery requested support for its nursing cohort of ~800 students in relation to their dosage calculations. A Drugulator prototype was presented to the faculty firstly as a means for students to generate the correct answer (to check their competence using traditional pen/paper/calculator methods) but also as a way for students to recognize a common principle underpinning all their calculations. With the prototype approved, there followed a period of development and refinement using feedback from faculty staff, local website design companies and students. In early 2007, a developer was hired to measure server performance and develop tools to monitor real-time student usage of the system. In the second semester of 2007 the site and interface was released to VU's nursing cohort.

It was recognized fairly early on that apart from this pedagogical application, allowing users to more efficiently and accurately calculate drug dosages has clinical benefits if it was readily available in the clinical environment. A business case study was then commissioned by VU to compare cost savings from potential ADE reductions with the cost of developing a customized hand-held product. As of early 2009 the University is seeking partners as part of the final round of IP protection[25].

The Drugulator is designed to be used by any health-care professional needing an efficient tool to compute patient drug dosages. At the point of care, such a tool would therefore currently be most useful for practicing nurses. Away from the point of care and, in particular, where users require dosage calculations involving a large number of variables, physicians, pharmacists, drug researchers can usefully apply the system to systematically automate a wide range of drug calculations. Students in any of the aforementioned areas would also benefit from learning to operate the Drugulator given that such operation requires and demonstrates an understanding of a common principle underpinning a wide variety of possible dosage calculations.

2.2. Overview & Illustration

In this initial description it is intended to convey the standard work-flow and a sense of the underlying *Constructor* principle that forms the basis of the Drugulator operation. Such a sense should be sufficient for our later discussion of the Drugulator method in relation to general FCAD and CPOE interfaces. For complete details however, of both the mathematical manipulations and the roles played by each of the interfaces’ components, readers should consult the Appendix. Note that the Appendix also contains a range of worked examples as part of demonstrating the interface’s claimed versatility. Finally, an even more comprehensive list of the dosage calculations that are possible using this principle can be found within the online Drugulator resource[1].

The Drugulator interface is shown in Figure 1 and consists of two separate interfaces. The left-hand side (L.H.S.) interface is used for any preliminary stock concentration calculations. It operates on more or less the same principle as the right-hand side (R.H.S.) interface which is the interface that ultimately computes the patient's dose, and hence the interface on which we shall focus.

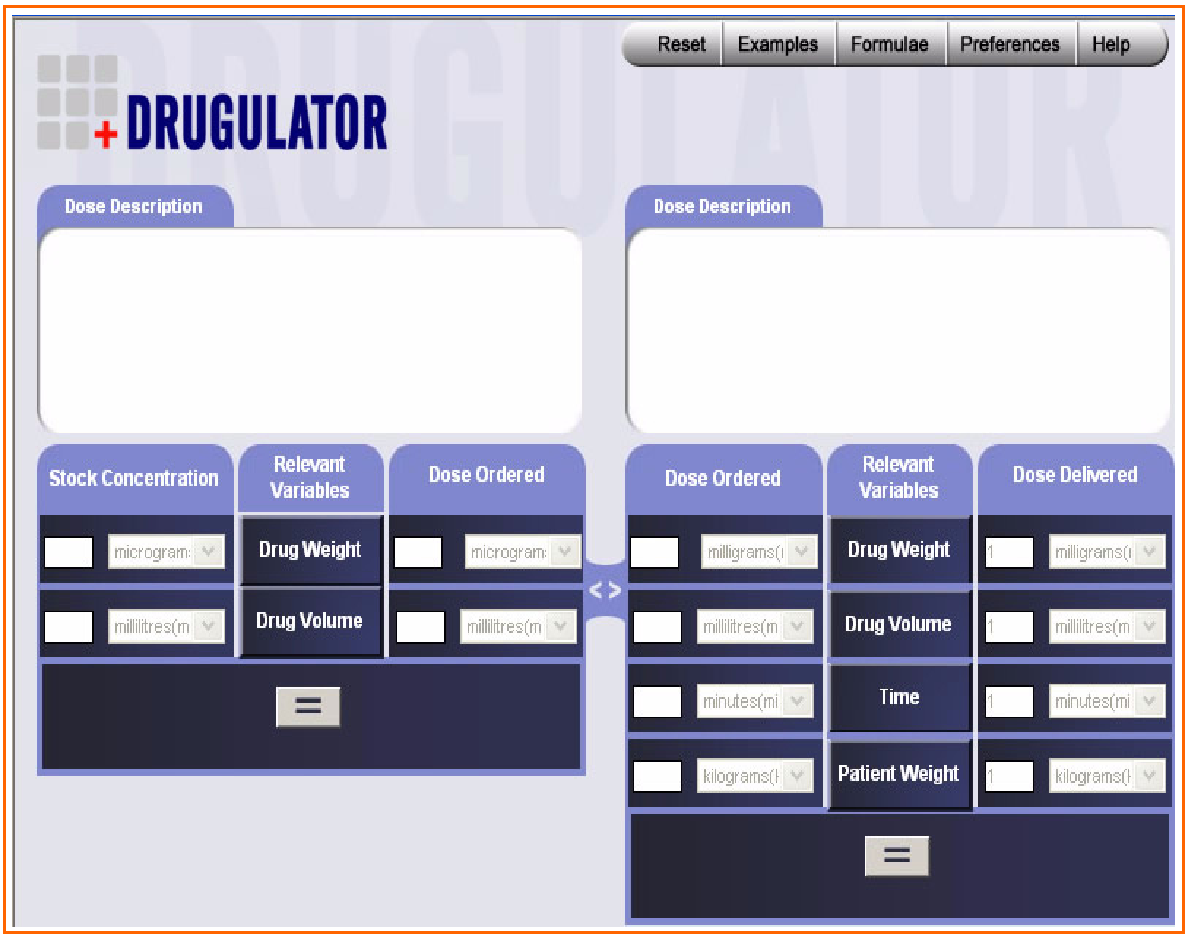
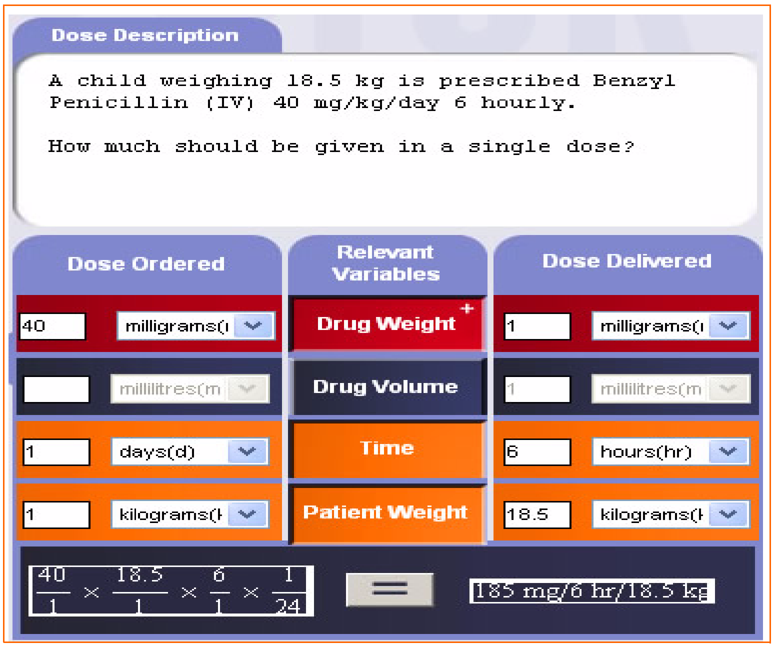


Figure 1: The Drugulator Interface before starting a drug calculation

The basic idea behind the interface is that users effectively *construct* a mathematical expression corresponding to the required dose by identifying three main components of a dose's calculation. The first component is simply those variables involved in the dose - with one variable distinguished as that corresponding to the required amount. The second component to be identified is the "base dose" that has been ordered for the patient's condition. The third component relates to how this base dose is modified according to the particular patient. Once these three components have been expressed in the interface, clicking the "=" button generates the constructed expression along with its evaluation corresponding to the final dose amount.

In Figure 2 therefore, the three components to be identified in calculating the correct dose (from information in the top *Dose Description* panel) are expressed in the interface as follows:



Dose Description

A child weighing 18.5 kg is prescribed Benzyl Penicillin (IV) 40 mg/kg/day 6 hourly.

How much should be given in a single dose?

Dose Ordered		Relevant Variables	Dose Delivered	
40	milligrams(	Drug Weight +	1	milligrams(
	millilitres(m	Drug Volume	1	millilitres(m
1	days(d)	Time	6	hours(hr)
1	kilograms(	Patient Weight	18.5	kilograms(

$\frac{40}{1} \times \frac{18.5}{1} \times \frac{6}{1} \times \frac{1}{24}$ = 185 mg/6 hr/18.5 kg

Figure 2: A weight-based calculation using the Drugulator's R.H.S. sub-interface.

The first component – identifying the relevant variable – is expressed by the colouring of the three rows; *Drug Weight, Time* and *Patient Weight* with the *Drug Weight* variable distinguished (with a red colouring given it is this variable's value that is being sought). The second component, the base dose, is a 40 mg/kg/day prescription and is expressed in the first column – the *Dose Ordered* panel. The final component, the information specific to this patient's administration appears in the second column - the *Dose Delivered* panel. With these features in place, clicking the "=" button causes the constructed mathematical expression to appear to the left of the "=" sign along with the numerical value it evaluates to, on the right hand side of this "=" sign.

Currently in this interface the constructed expression reflects a *proportional* relationship that is assumed to exist between the dose variable and the other variables on which it depends (note also that any unit conversions are handled automatically). Such a relationship, perhaps surprisingly, includes the vast majority of dosages required of nurses and hence needed by any putative FCAD[1]. This coverage is what gives the interface its general utility and evidence for this claim, as mentioned earlier, is provided in the Appendix.

2.3. Performance Metrics

The main design objective of the Drugulator to be discussed here relates to its viability as a calculating tool at the point of care. This viability specifically relates to its efficiency over a sufficiently wide range of required dosage calculations. Currently there does not appear to be any standard measure of this viability and indeed, one of the motivations of this paper is to highlight the need for more comprehensive studies into the effectiveness of FCADs being considered for clinical application. To this end we introduce several measures as a way of providing an initial evaluation (with each subsequently applied to the Drugulator). Initial evaluations can be a useful filter to identify unviable systems but are clearly not sufficient for any mooted clinical adoption. For this, comprehensive, longitudinal studies over a variety of situations and with large numbers of users are needed. Such trials have *not* been carried out on the Drugulator although in what follows we describe several design features that, we argue, provide natural advantages in relation to the developed metrics.

2.3.1. A FCAD Categorization

The main performance metric(s) will be framed around three different FCAD categories - *Selectors*, *Constructors* and *Open-Enders* (see Figure 3). This categorization is useful because it distinguishes between current practice, existing automated systems and the Drugulator method. Each category also defines qualitatively different FCAD user-experiences which will be reflected in different scores on the defined metrics.

[1] This fact greatly simplifies the underlying implementation although such a relationship is not an intrinsic feature of the method: For example, later we show how it can be extended to other variables/ relationships and hence as part of a comprehensive CPOE$_{cds}$.

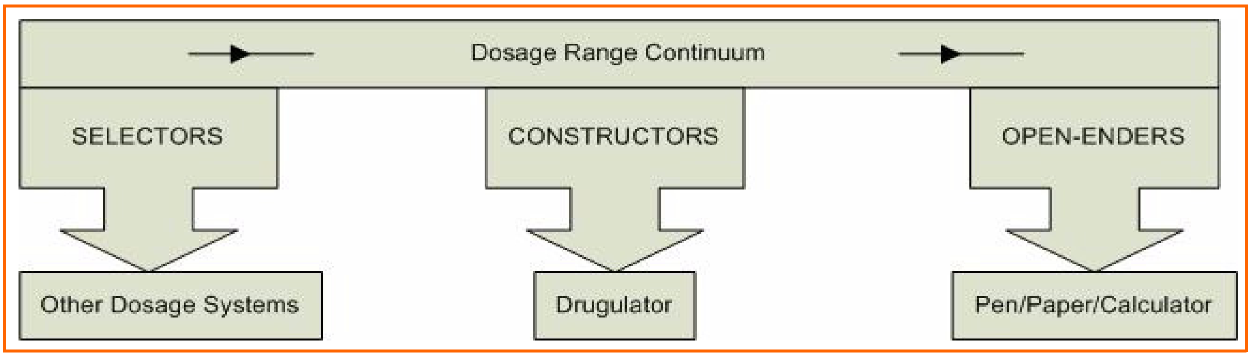


Figure 3 : Categorizing a FCAD as either a- *Selector*, *Constructor* or *Open-Ender*

Selectors, the most common type of dosage system, are categorized by having a separate interface for each possible dosage calculation. *Constructors*, of which the Drugulator is an initial implementation, are distinguished by the way in which dosages are *constructed* from the relevant dose variables. *Open-Enders* incorporate the current method whereby essentially *any* calculation can be performed and hence, in particular, any possible dosage calculation. We now discuss three different metrics in relation to systems defined in each of these categories.

2.3.2. Coverage

The *coverage* of any FCAD - the range of possible calculations it is able to perform - is a basic part of any system's evaluation and hence, we argue, should be the first step in any systematic analysis. We frame the analysis of coverage here around the previous categorization but briefly note the following: The coverage of *Selectors* is finite (essentially equal to the number of different interfaces in the system) but, as we will illustrate, is of an order of magnitude less than the coverage of *Constructors* (which is also finite). The coverage of *Open-Enders* is infinite.

2.3.3. Open-Enders

In principle, the use of pen/paper/calculator allows the health practitioner to perform any possible dosage calculation. While obviously useful, such freedom also includes the freedom to commit any type of error. Furthermore, dosage calculations contain certain repetitive components through which efficiencies can be gained by collecting them into a computer interface. Finally, there are other reasons related to automated checking and links to decision-support systems that, when combined with persistently high ADE rates, makes a compelling case to consider ending this method of dosage calculation in the clinical setting. In most of what is to follow we will be more concerned with comparing *Selectors* with *Constructors*.

2.3.4. Constructors

In *Constructors* users construct a mathematical expression corresponding to the required dosage calculation by first "activating" those variables involved in the calculation. Since there can only ever be a finite number of variables in the system at any one time, the number of possible dosage calculations that can be performed by a constructor system is therefore also finite[1]. The freedom however, of being able to *choose* the relevant variables - amongst say n different variables - means that an exponential number, 2^n of different dosage calculations becomes possible. Mathematically, this is simply the size of the *power* set of n elements but in the Drugulator context can be observed by noting that each row can either be in the "on" or "off" position depending on whether or not the corresponding variable is activated. The total number of ways each of n rows can be either "on" or "off" is 2^n.

While the freedom of choosing variables allows any calculation using such variables to be constructed, the constraint associated with the way in which the system forms the corresponding mathematical expression, means that there is less opportunity for non-dosage computations to be formed. Currently (as detailed in the Appendix), the way in which the system forms the corresponding mathematical expression assumes a *proportional* relationship between independent variables and the final calculated dosage. It turns out this is sufficient for the vast majority of point of care drug calculations although this proportional relationship, is not an inherent feature of constructor method within a FCAD. Indeed, other relationships can be readily augmented with the addition of new variables.

The Drugulator method, as far as we are aware, is the only dosage calculator or FCAD that employs the constructor method and we can, based on the previous argument, use it to estimate its coverage. There are two interfaces to consider. Firstly, the L.H.S. Stock Concentration interface

[1] This finiteness refers to a “frozen” version or a particular FCAD instance. Any FCAD’s range can, of course, always be extended via its ongoing maturation (which for *Constructors* would mean the addition of new variables).

has 2 different configurations depending on which of the two variables is identified as being dependent (thereby consigning the remaining variable to be independent). Secondly, the R.H.S. interface specifies 4 variables giving 2^4 different ways these can be activated. A preliminary stock concentration is either required or not giving a final coverage of $2 \times 2 \times 2^4 = 64$. In practice, the actual number of calculations will be somewhat less than this due to certain administering constraints associated with dosing practice in general (e.g. the dependent variable is most likely to be either *Drug Weight* or *Drug Volume* and both are unlikely to ever be simultaneously activated). Hence, 64 is the *maximum* number of different dosage calculations that involve proportionate relationships between dependent and independent variables drawn from *Drug Weight, Drug Volume, Time* and *Patient Weight* (including a possible preliminary stock concentration calculation)[1].

2.4. Selectors

The selector approach invariably used by current dosage calculating systems, involves setting up a separate interface for a separate dosage calculation. This interface can then be used when the corresponding calculation is required. Hence the coverage of such systems is equal to the number of such interfaces so constructed - a value which invariably reflects the system's maturity. The selector approach is therefore potentially useful when the total number of different interfaces is low and the interfaces contain a large number of specific and unusual variables. Conversely, when common variables are repeatedly combined to produce the final dosage calculation, the system's coverage becomes exponential in the number of possible variables. That is, n different variables give rise to 2^n possible dosage calculations which then require, under a selector method, 2^n different interfaces.

2.4.1. Comparing Coverages

In principle, both *Constructors* and *Selectors* can be designed to cover the same range of calculations. The fundamental difference between the two methods however, becomes apparent once their respective usability over the entire range of possible dosage calculations is considered[2]. This will be reflected in the *time* taken to navigate amongst this range but this time can also be directly estimated with another measure specifically related to a system's coverage - namely its *Efficiency Index*. Before defining this measure, we motivate its definition by recalling the most immediate manifestation of employing a *Constructor* or a *Selector* - namely the number of resulting interfaces. Collectively reviewing the examples in the Appendix (or a more extended range from its online manual) shows that many of the Drugulator's input fields are being repeatedly re-used. In effect, common types of arguments are being "extracted" from various functions that each implement a different dosage calculation. This extraction means that similar types of arguments are not being repeated each time the function and its corresponding dosage needs to be calculated. For example, *Time* is involved in every dosage calculation that involves some sort of *rate*. In the Drugulator's single interface, instead of an input field corresponding to *Time* being repeated for every such calculation, it appears only once and hence can be activated from the one interface when required.

2.4.2. Efficiency Index

The aforementioned description captures how "extracting common arguments" can result in a single interface and the associated usability advantages. On the other hand, it could be argued that any other dosage system employing the selector approach (i.e. with 1-interface per 1-calculation), could in principle, also employ a single interface simply by concatenating all of its interfaces into one "super-interface". Naturally such a super-interface would have serious usability issues both in terms of users finding the appropriate "sub-interface" (in effect, the aforementioned search is now being conducted on a single screen) as well as navigating amongst all the necessarily compressed input fields. The above two examples motivate the following Efficiency Index (EI) definition for FCADs. The definition aims to measure and distinguish between, the efficiency gained from re-using arguments as in the constructor model and the lack of efficiency in the 1-interface 1-calculation paradigm characterized by *Selectors*.

$$EI(Sys) = \frac{Coverage(Sys)}{\#\text{ Inputs}}.$$

Hence, the Efficiency Index of a FCAD system measures the number of different calculations per input field. In the Drugulator interface, for example, there are 24 inputs (4 inputs for each of the 6 rows that in total make up the L.H.S. and R.H.S interfaces) so that a coverage of 64 gives an EI (Drugulator) of (64/24) ~ 2.67. For a Selector, employing a 1-interface 1-calculation approach, the EI reduces to the reciprocal of the average number of inputs per calculation. Estimating this average at ~6 we have an EI estimate of 1/6 = 0.2 for a Selector (that is invariant to concatenation). Note that for arbitrary n, we have EI(Constructor) ~ $O(2^n)$ whereas EI(Selector)~1. Hence we see that this index measures the efficiency with which input fields are re-used and is, as we will argue, ultimately correlated with the time it takes for users to perform a typical dosage calculation - an issue to which we now turn.

[1] Note that implicit in discussions of the Drugulator's coverage is that dosage calculations differing only in the value of the respective inputs are still considered to be the same. For example, changing a quantity's units is not counted as changing the type of calculation.

[2] One other significant difference is the ease of implementation between the two types of systems. Since a *Constructor* involves grouping variables that are related in similar ways, this relationship need only be implemented once. For example, activated variables in the Drugulator interface define a proportional relationship between variables and hence this proportionality, once implemented, can be used to cover a variety of situations.

2.4.3. Time

Once a dosage system has sufficient coverage, time-pressured health-care professionals need to be able to generate a dose from within this coverage in a "reasonable" amount of time. This is especially so given the aforementioned and well-documented sensitivity to this precious resource. As a first approximation this capacity should be measured over *all* doses. Specifically, a *Time* measure can be set to be the average time taken to perform a dosage calculation over all the possible dosage calculations that make up the system's coverage. This time consists of 4 parts: The time; (1) to locate/construct the correct dosage calculation (2) to input the relevant parameter values (3) to perform the mechanical evaluation and (4) to perform any final checks. Since components (2), (3) and (4), depend on factors that are essentially independent of whether a FCAD is a *Selector* or *Constructor*, it is component (1) that can be used to differentiate a FCAD's efficiency and used as a first approximation when evaluating respective feasibilities.

2.4.4. Comparing Constructors & Selectors

It is when considering the time taken for a user to construct or locate the relevant dosage calculation in either a *Constructor* or a *Selector* that a fundamental difference begins to emerge. A *Constructor*, consisting of a single interface, allows users to focus on constructing a formula by identifying those variables relevant to the dosage calculation. Conversely, users on a *Selector*, having performed a similar variable identification, then need to also find the correct interface from amongst its existing repertoire. It is precisely this additional selection time, we argue, that has prevented FCADs from becoming a viable alternative to current practices in the clinical setting.

In the Drugulator embodiment of a *Constructor* FCAD, it is estimated that a proficient user should be able to identify the relevant variables within about 10 seconds. In a *Selector* FCAD, users also have to identify the relevant variables but then on top of this, *find* the appropriate interface that employs such variables in the correct way. How long this takes will depend on the number of interfaces in the FCAD (and how well organized they are). Even for a small number of variables however (e.g. those used to define the Drugulator's coverage), we argue that this time is significant for a *Selector* containing a similar coverage. That is, we argue that a *Selector* with a search of time of anywhere between 30 to 40s can start to affect a FCAD's practical usability given that the *total* time also includes the aforementioned parts (2), (3), (4) along with the time for the physical administration. What is less arguable however, is how these practical differences become magnified once additional variables are augmented in the system (e.g. as part of an integration with a CPOE_{cds}). More precisely; as previously observed, the coverage of *Selectors*, being equivalent to the number of interfaces it contains, increases exponentially in the number of variables compared with a linear increase associated with *Constructors*.

2.4.5. Correctness

Once a system has adequate *Coverage* and a feasible *Time* metric value, the final component needed to evaluate the system's effectiveness is how often users generate the correct dosage. Again, as with the prior two metrics, this metric should be a measure over all the doses within the system. That is, a FCAD's *Correctness* measures the percentage of calculations that users get correct over the FCAD's coverage. One feature of *Constructors* that enhances their *Correctness* is that by effectively constructing the appropriate formulae through interactions with its interface, users need to bring to bear their mathematical/linguistic understandings in a similar way as they would using pen/paper/calculator. What this means is that users interact with the system in those areas that require human abilities - namely in the translation from a drug dose into a mathematical expression. Compare this with the inherent passivity of *selecting* an appropriate dosage calculation from within a *Selector*. This passivity contains two inherent risks; the first involves selecting a similar but incorrect interface while the second arises in the case of *no* appropriate interface being available. In this latter case there is the inevitable temptation to co-opt an existing but inappropriate interface. Finally, it is also worth noting the potential increases in correctness common to the automation inherent in both *Constructor* and *Selector* FCADs; namely the mechanical evaluation of a formula together with the opportunity to cross-check with safe dosage-ranges from digital pharmacopoeias.

One feature, implemented in the Drugulator, but not related to its constructor nature, is that any generated dosage is always accompanied by the mathematical expression used to generate it (The expression to the left of the "=" sign in the interface). This can be an indispensable aid for checking since it confirms to the user, the underlying algorithm being used by the system. Given the lack of user-input in constructing the corresponding mathematical expression in existing *Selectors*, such a check would presumably assume greater significance in these systems although somewhat curiously, such a feature appears to be very seldom implemented.

2.4.6. Summary

In this section we introduced an *Efficiency Index* which, together with other measures - *Coverage, Time* and *Correctness* - can be used to evaluate a FCAD's potential usability. We have used these metrics in relation to the Drugulator, a type of constructor FCAD, and argued that using a selector-based FCAD quickly comes infeasible as additional factors are incorporated - as occurs with dosing of increasing sophistication. The Drugulator method can be thought of as lying half-way between the extreme of having complete freedom to compute anything (pen/paper/calculator) and the other extreme of only being able to compute those dosages that apriori have been added to the system. The net effect of this "betwixtness" is that a human's input is being emphasized where it is most needed - in constructing a formulae from a clinical situation - whilst maintaining the capacity to de-emphasise it where it is least needed - in the formula's mechanical evaluation.

3. Extensions/Relations

3.1. Constructor *CPOE* Systems

The previously developed metrics aimed to evaluate FCADs as part of the *Administer* stage in the process of medical management. The main principle underlying constructors can however, also be applied to CPOE$_{cds}$s as part of the first *Prescribe* stage. Indeed improvements in usability provided by the constructor idiom, are potentially more pronounced in a CPOE$_{cds}$ system given that these improvements are compounded each time a new variable is added.

For example, even incorporating one extra variable - *Patient Weight* - is correlated with elevated levels of both medication errors and ADE rates[14, 34] within paediatric dosing and intensive care units. To be sure, this correlation between weight-based paediatric dosing and ADE rates does not reflect a single causal link given other risk factors such as narrower "therapeutic windows" for young children as well as a reduced capacity to communicate adverse reactions. Nonetheless, the complexity of a dose is a significant factor in error rates and in the case of patient weight this has been indicated even for non-paediatric situations[35, 36]. The potential effects on ADE rates therefore, is potentially much greater when including some or all of the following factors: Patient Weight, Age, Surface Area, Renal Function, Liver Function, Drug-Allergy, Drug-Drug Interaction and even emerging DNA individualization. Most of these factors have been identified as being integral to a comprehensive CPOE$_{cds}$ application [26, 37] and can also be potentially applied in defining individual biochemical profiles as part of personalized dosing [38].

Clearly applying the constructor principle to CPOE systems would lead to adjustments in the underlying algorithms depending on what additional variables are activated as part of the calculated dosage - effectively integrating a form of clinical decision support. Whatever the adjustments, our experience with the Drugulator, suggests that any applied algorithm be made as transparent as possible (admittedly this is more straightforward in the Drugulator since it is an arithmetical, proportional relationships that are being communicated). Users appreciated knowing exactly how an answer was obtained since it increased both their understanding of, and their confidence in, the system. While this would be more difficult to implement in a CPOE (given that decision rules effectively need to be communicated) we argue that similar levels of proficiency and confidence would be inspired by a similar levels of transparency. Such transparency requires developing an appropriate terminology [39] but ultimately enables a final human check or else a confident override in the case of a “false alarm” - a common problem amongst existing CPOE$_{cds}$s [26].

One feature of CPOE$_{cds}$s that nicely crystallizes the distinction between *Constructors* and *Selectors* is the use of so-called *quick orders* or their collective counterpart, *Order Sets* (collections of ordered quick orders). In a *quick order*, input fields are predefined and usually filled in with (editable) default values so that their application employs a process in which physicians *select* a particular order amongst a larger set of possible dosages. Adding such a feature to CPOE$_{cds}$s has shown some promise [19] [20] [40] mainly because immediately selecting a dose can, in standard situations, be more efficient as well as reducing the likelihood of transcription or inadvertent errors. Such an approach however, will always be fundamentally limited by the large number of *quick orders* that must inevitably result from doses formed by combinations of several variables. That is, once the number of order sets becomes sufficiently large, the accompanying search problem will inevitably and rapidly lead to a *general* loss of efficiency. This was noted earlier in this section and the limitation has been readily observed in the CPOE$_{cds}$ context [41] [42] (where, for example,one CPOE$_{cds}$ system contained 7423 quick orders [41]).

Others have reported a great deal of effort spent developing hundreds of order sets only to later realize that personal order sets were neither valued nor often used (at only 13% of the time [41]). In addition to the time spent on the initial build of the order sets, finding the correct one in the system, and routine maintenance or updates can quickly become overwhelming [20].

There have been some approaches to managing large numbers of order sets by restricting the search to those sets judged most applicable for any given patient. Determining these "most likely" order sets can be done, for example, by using patient's admission diagnosis, ward location or clinical service [20] and can also include a probabilistic analysis of such factors [40] or even the application of neural networks [43]. The constructor approach illustrated in the Drugulator provides an alternative method of efficiently navigating this large "order-space". It allows physicians to generate an order by first selecting the order's relevant variables while simultaneously requiring them to apply their human judgment where it is appropriately required.

It is worth noting that the aforementioned advantages and accrued knowledge associated with implementing Order Sets in CPOE$_{cds}$ systems can still be usefully transferred to a *Constructor*. For example, the menu categorizations created for Order Sets has a "Variable Sets" analogue within *Constructors* (which should also be made customizable in the Drugulator as it is in CPOE$_{cds}$ systems [44]) while recurring default values in the input fields of selected variables would continue to act as time-savers. While these would improve the Drugulator and any other *Constructor*, it is important that the flexibility of being able to "activate" a relevant variable and change its corresponding input fields be maintained lest the system slip back into becoming a *Selector*.

The sheer complexity of the medical dosing pathway ensures that the design of CPOE$_{cds}$ systems, assisting even the first *Prescribe* stage, will inevitably be of considerable complexity and face ongoing challenges [45] [46]. Indeed, as of 2007, “designing a complex yet flexible, protocol-based, safe dosing approach to chemotherapy remains a relatively unsolved problem in most institutions implementing CDS within CPOE” [19]. One important aspect of this design includes physicians being able to digitize orders in a timely manner. The constructor principle implemented in the Drugulator allows computer-aided dosage calculation in the *Administer* stage of this process which, as we have just argued, can be extended to provide similar efficiencies in a CPOE$_{cds}$ system focussing on the *Prescribe* stage. In whatever way the dosing problem is eventually "solved", a degree of uniformity between both FCAD and CPOE$_{cds}$ systems as a way of connecting more closely the *Prescribe* and *Administer* stages of the medication process, would be beneficial.

3.2. Pharmodynamics & Pharmacokinetics

Advances in the detail and sophistication of mathematically modelling in both pharmodynamics and pharmacokinetics come with the associated computational problem of transferring such findings into practical clinical orders that can be efficiently generated by time-pressured healthcare professionals. The simplicity, uniformity and extensibility of *Constructors* provide a technological bridge across which more sophisticated, and possibly more efficacious, dosage regimes can be implemented. That is, using a similar *Constructor* system to collate and manage increasing number of dosage factors as part of pharmodynamic and pharmacokinetic research, potentially allows a more seamless transition between outcomes of this research and their eventual clinical application. In particular, having the same apparatus for both domains reduces the likelihood of discovering efficacious new dosage regimes, which are nonetheless too complex to be usefully and routinely applied in the clinical setting. Conversely, this uniformity means (suitably anonymised) usage data can be fed back into this research program, and more specifically, contribute to a standardization that has been argued as being critical in uncovering ADE origins [47] [48].

3.3. Competency and Pedagogy

The reader will have noted that of the discussed measures *Coverage*, *Time*, and *Correctness*, inherent in the last two is a subjective evaluation of a user's competence in driving any tested FCAD. This competence is, in turn, also related to the effectiveness of any accompanying training packages which can also be used to establish benchmarks that more rigorously establish a FCAD's effectiveness. In relation to the Drugulator, its website currently contains a user manual although currently no inbuilt assessment system yet exists to establish such benchmarks. Naturally such benchmarks can also help establish, to a much higher degree than has previously been possible, the competency of health-care professionals in the area of dosage calculations.

In this paper the Drugulator is being put forward as representing a possible paradigm shift in the way dosage calculations are performed in the clinical environment. Learning to operate the Drugulator however, requires learning a principle that underpins all nursing calculations and hence can still be of benefit to students learning dosage calculations with traditional methods. It is in this capacity that the Drugulator is currently being used at Victoria University and which we briefly describe.

Recall that the vast majority of dosage calculations expected of nurses are based on the straightforward concept of a dependent variable changing *in proportion* to one or more other independent variable(s). Unfortunately, this concept is often obscured by a variety of factors that make these calculations more complicated than they really need to be. Some of these factors include: an over-reliance on formulae in education; the need to perform the arithmetic by hand or calculator; the additional bookkeeping imposed from multiple independent variables along with any unit conversions; and finally, the need to sometimes perform preliminary stock concentration calculations. By applying the Drugulator and in particular by automating many of these obscuring factors, the conceptual essence of the calculation crystallizes: the identification of the *in-proportion* relationship between the relevant dependent and independent variable(s).

3.4. Lessons Learned

As previously illustrated, the initial Drugulator interface was implemented within an internet web-browser (also see Figure 12). As development and trials proceeded however, it became clear that interface interaction was every bit as important as the structure of output generated from the attached computational back-end. Web-browsers however, do not contain the same flexibility and features of other environments optimized for interface development. Hence, while the resulting web-presence is clearly useful from a dissemination viewpoint (enabling users to, at least, get a feel for the Drugulator's operation), a clinical application would benefit from one of these more powerful interface environments.

In the Drugulator development, while recognizing and implementing an “all-purpose” mathematical algorithm was an important starting point, it quickly became apparent that a wider range of expertise would be necessary to capitalize on the opportunities arising from any introduced automation. In particular, as part of any possible systemic change it became clear that any truly effective FCAD, would require sustained input from nurses, physicians, specialists in medical informatics, information technologists, graphic designers, educators, software engineers, usability experts and designers of both digital pharmacopoeias and CPOE$_{cds}$s.

The importance of an integrated, multi-disciplinary approach has often been noted[6] [8] [46] but remains an elusive goal mainly due to the high upfront costs associated with the necessary technological and cultural convergence. In a later section, *A Developmental Blueprint*, it is shown how the Drugulator’s design (and that of any *Constructor*) can address this via an inbuilt extensibility that could be further enhanced by the development of more refined FCAD and CPOE *standards*. From our experience with the Drugulator, the following two factors represent the most significant in relation to any potential FCAD standards. Firstly, any internal computations or algorithms should always be made as transparent as possible to the user and secondly, that a *Constructor* idiom be adopted as a way of allowing users to construct a patient’s dose.

3.5. Alternative Implementations

Thus far, the term *The Drugulator* has been used somewhat interchangeably referring sometimes to the online interface and sometimes to the website. Most accurately however, it is a *method* for performing dosage calculations characterized by the previously discussed *constructor* idiom. The interface and technical infrastructure as described in this paper represent a first, working implementation of this method - but others are possible and probably more desirable.

A discussion of the design principles underpinning of the Drugulator's interface was deferred to emphasise the primacy of first developing metrics to evaluating the *performance* of any implemented dosage-calculator. In line with this, some of the design features introduced in the Drugulator remain to be fully tested with the objectivity of such metrics. There are however, some central questions concerning the optimality of particular features that arise from the adoption of *Constructors* - many of which are currently being investigated. Such questions about the Drugulator design include but are not limited to: What is the optimal arrangement of the selectable variables? (static or dynamic order, permanently visible or accessible via a menu); How should the Drugulator variables most easily be selected? (using click order or via a menu); How should the Stock Concentration interface be integrated with the Drugulator? (permanently visible, via a pop-up, with outputs automatically inserted into Drugulator fields); What is the most seamless method for transferring Drug Order data into the Drugulator interface? (simulating Drug Chart templates etc); How should the internal algorithms be most clearly communicated to the user? (variations in arithmetic form, real-time updates, natural language explanations).

An important decision affecting clinical practice with the Drugulator concerns access from either a standard computer screen or via a hand-held device. The easiest path both logistically and technically is via a single computer screen (for example an internet-connected workstation can now access this Drugulator implementation). In an initial pilot study however, ready access to these in-demand workstations became an issue in the clinical setting so that migration to an individual hand-held device is most likely preferable. Such a migration raises several questions: what design changes are needed for a smaller screen? Should the enabling software reside on the hand-held device or within a central location accessed wirelessly? How can the integrity and security of any wirelessly transmitted information be guaranteed? Should the hand-held device leverage current PDA technologies or become a custom-made medical application? While these questions represent significant design and technical challenges, none appear inherently intractable and are currently being investigated as part of the Drugulator's development.

The current Drugulator implementation adopts a *client-server* architecture in which a single web server is being used to distribute dosage calculations to any number of clients (Internet Explorer browsers). An alternative architecture would be to posit the application entirely on the user's individual machine or handheld device. Both types of architecture have advantages/disadvantages although there are particular clinical imperatives that probably make the client-server model more appropriate for a comprehensive FCAD. One of the main advantages of the client-server model is that any upgrades to the system's logic can occur centrally and from the user’s perspective, unobtrusively and without any mass distribution of software upgrades. In a FCAD such upgrades assume special significance in relation to safety given envisaged integration with large digitized pharmacopoeias and/or more complex dosage calculating algorithms.

The other opportunity offered by a client-server architecture is the ability to record all usage of the system by health-care professionals. While this inevitably raises privacy issues, once these are judiciously managed, broad opportunities exist for continuous systemic improvement. For example, the usage data can yield immediate electronic trails for ADE audits and evaluations of the systems performance in relation to the metrics discussed earlier. Finally, while a client-server implementation in the clinical situation would most likely posit the server at the hospital level, the web-based, client-server implementation allows a ready dissemination of its method as well as, educationally, being a reasonably cost-effective way of reaching large number of students.

3.6. Developmental Blueprint

The optimal, relative emphases of human input verses automation in health-care ultimately depends on preserving those tasks that require human judgment while developing systems to manage the more mechanical, repetitive tasks. In relation to the medication process, the initial human judgment required to compose a dose is typically followed by a series of steps, many of which are of an algorithmic nature and therefore amenable to advantageous automation. Currently, any automation within these steps is implemented on a somewhat ad-hoc basis and consists to varying degrees of, for example: CPOE$_{cds}$s; hand-held calculators; PDAs; IV Pumps and more recently, bar-coding and robotic dispensing. The automation introduced by the Drugulator involves the construction of a dosage's mathematical formulation in the *Administer* stage. It also however, creates a pathway for further incremental automation associated with each of the variables used in the formulation. The following is an illustration of a possible endpoint for this pathway in relation to a FCAD in the *Administer* stage (compare with a suggested automation across the entire medication process [49]).

3.6.1. A Hypothetical Scenario

Nurse Betty is alerted by her Drugulator beeping that Mary - one of her patients - is now due for her next dose. She then activates her Drugulator's "medical records" module to download any physician updates and/or details relevant to Mary's base dose (including adjustments due to her body weight and dose schedule). Betty checks the Drugulator screen to confirm that indeed the correct values have been imported into the interface's *Patient Weight* and *Time* bands. She then activates the Drugulator's "inventory module" to obtain the location of the nearest drug cabinet containing the required stock. After arriving at the cabinet Betty then attaches her Drugulator to the designated bottle and activates the Drugulator's "scanning module" on the bottle's barcode to confirm the correct drug selection. She then looks at the screen to ensure that the correct concentration values have been imported into its *Stock Concentration* interface (and in particular, its *Drug Weight* and *Drug Volume* bands). Satisfied with this, Betty then triggers her Drugulator's "extraction module" to extract the required drug amount before this is automatically transferred to the attached "delivery module" - in this case a syringe. Betty then proceeds to Mary's bedside before again activating the "scanning module" on Mary's barcoded wristband to confirm her identity. She then checks the Drugulator's screen to confirm both the dose's mathematical expression and the reasonableness of the evaluated answer, before performing a more comprehensive final check by activating (through a wireless connection) her Drugulator's "drug pharmacopeia" and "medical records" modules. Finally, with all these checks passed, Betty activates the drug-delivery release module and administers the injection upon which the relevant modules wirelessly update Mary's medical records and the hospital inventory.

Such a scenario although hypothetical is also fairly natural to envisage given the relevant technological developments and indeed, in different guises, has been the subject of numerous patents [50] [51] [37] [50] [53][54][55][43][56][57][38]. It is also now being pursued as part of the Drugulator's commercialisation. Its ultimate feasibility however, rests crucially on the *usability* and *extensibility* of any implemented interface – both of which, we argue, are addressed by a constructor model. The usability advantages in terms of reducing the time taken to calculate a dose have already been discussed but equally important is an inherent extensibility to allow for future changes in dosage regimes and differing rates of technological development. In particular, the rationale behind the Drugulator, or any *Constructor*, being the core interface through which a pathway towards the described automation can be achieved, rests on its focus on a calculation's relevant variables. Each variable contributing to a final dose represents an opening for human error and therefore also an opportunity for improving safety by the automatic *and* human checking of such inputs. It represents a *pathway* rather than an instant solution because the automation can be *progressively* implemented and not rely on a technological confluence that requires a simultaneous activation of all the respective modules. For example, a deployment in one setting may have wireless connections to digitized patient records but still require the manual inputting of drug concentration data until such time as all its storage bottles have been bar-coded. In another health care system, in the developing world for example, there may be no opportunities to automate any inputting so that all inputs need to be manually added as described in this paper.

4. Conclusion

Finally we collate and summarize what we see as the key advantages and possible drawbacks of the Drugulator system.

4.1. Advantages

Realising Automation: Automating any component of the medication process offers the promise of greater correctness, efficiency and ultimately safety particularly for those components that are sufficiently “mechanical”. Drug calculations especially their construction and evaluation, clearly possesses this mechanical nature and hence are amenable to future automation. Any potential automation however, is still conditional on any implementation attaining a high degree of usability. This is because at some point in the medication process, human judgement is still required in any system interaction. The rationale behind the Drugulator method is that, in the face of a diverse and possibly increasing range of possible drug calculations, the type of human interaction facilitated by its single interface can, and perhaps uniquely so, allow this level of usability to be reached.

Progressive Improvement: There exists a clear entry barrier related to the level of development required prior to clinical application and this is identified as such in the next *Disadvantages* section. With this barrier surmounted however, the Drugulator method is also “future proof” in two important senses. Firstly it can be progressively connected to other medical systems while secondly; its claimed usability is not compromised by an inevitable increase in drug-calculating sophistication and precision.

4.2. Disadvantages

Quantum of Change: One factor militating against an eventual adoption of the Drugulator method is that it is, in some ways, a radical departure from current methods. Consequently, there is a large quantum of development, implementation and attitudinal change required to overcome long-standing and entrenched practice. Reaching such a point may not be achievable organically in small incremental steps but instead, require a large injection of resources and institutional commitment typically predicated on widespread, expensive trials.

Steepness of Learning Curve: The Drugulator, like any tool, requires a period of familiarization and training before its efficiency and accuracy gains start to become realised. In this initial implementation, a training period of a few hours was sufficient before most learners reached competence. This however, was based on instruction been provided by the developers and hence could possibly be greater within an uninitiated environment.

Risks from Incorrect Input: The Drugulator, like any automated process requires, at the some point in the medication process, correct input to obtain correct output. While, as discussed, this is potentially mitigated through automated checks with connected systems, there remains the risk of users driving the system without applying their mathematical sense either during its operation or in a final check.

Premature Dependence: When used as a pedagogical tool, the Drugulator removes the need for the manipulation and evaluation of any constructed formulae. Prior to any potential introduction into the clinical environment therefore, care needs to be taken to ensure that in any training, it is only used after, or in conjunction with, existing methods (to this end, in the training of nurses at VU, it is only used as an extension to the standard to pen/paper/calculator methods).

4.3. Concluding Remarks

This paper has introduced *The Drugulator* - a new method for performing drug calculations. The method can be used to underpin a computer system through which it is envisioned that health-care professionals can produce correct dosages with greater reliability and efficiency. An initial, web-based implementation of this method is presented and illustrated together with numerous examples. The Drugulator method itself, points to a certain *constructor* idiom having inherent advantages over a *selector* idiom that is currently being used in existing systems. Indeed we have argued that this new idiom not only makes current point-of-care calculations feasible, but that it becomes indispensable for any serious automation of more sophisticated dosage regimes. This would occur, for example in CPOE$_{cds}$s using increasing numbers of variables. Further, once such an idiom is adopted we have suggested how, through a progressive integration with existing hospital systems, it can be used as a pathway towards greater automation and reliability in drug calculation and delivery.

Whilst the online Drugulator model of this paper represents a working implementation, we have also urged that further testing be performed to evaluate not only the Drugulator's general performance but also any FCAD being put forward as an automated dosage calculating system in the *Administer* stage of medical management. Towards this end we have defined several metrics and shown how they can be used to capture a FCAD's overall performance. While the upfront cost of this putative testing and development of a pathway to increased automation in integrated FCAD and CPOE$_{cds}$s is obviously high; it would also seem to be significantly less than the enormous costs associated with the damage currently being inflicted through existing rates of ADEs.

Acknowledgements

Ms Geraldine Rebeiro and Dr Trish Burton from Victoria University's School of Nursing and Midwifery provided valuable feedback and encouragement throughout the Drugulator's development and also greatly

assisted in its introduction to VU's students. Dr Mike Honeychurch performed much of the server load testing as well as writing the *webMathematica* code to be used in measuring students' usage of the Drugulator. The implementation of the design specifications for the Drugulator web-interface was performed by the web-design company - EXA and funded by VU's Portfolio of Language and Learning. It is also disclosed here that the author, as per Victoria University's Intellectual Property Policy, is entitled to a proportion of any net revenue resulting from any commercial exploitation of the Drugulator technology.

References

1. Monson RC. Online Drugulator. The version referred to in this paper has been frozen and is accessible from http://drugulator.ronaldmonson.com. (2007) The official Drugulator site is at http://www.drugulator.com (2007) Currently, these versions are identical although they will possibly diverge over time.

2. Bates DW, Spell N, Cullen DJ, Burdick E, Laird N, Petersen LA, Small SD, Sweitzer BJ, Leape LL. The costs of adverse drug events in hospitalized patients. Adverse drug events prevention study group. Journal of the American Medical Association. 1997; 277(4): 307-11.

3. Johnson JA, Bootman JL. Drug-related morbidity and mortality. A cost-of-illness model. Archives of Internal Medicine. 1995; 155(18): 1949-56.

4. Leape LL, Brennan TA, Laird N, Lawthers AG, Localio AR, Barnes BA, Hebert L, Newhouse JP, Weiler PC, Hiatt H. The nature of adverse events in hospitalized patients. Results of the Harvard medical practice study ii. New England Journal of Medicine. 1991; 324(6): 377-84.

5. Lazarou J, Pomeranz BH, Corey PN. Incidence of adverse drug reactions in hospitalized patients: A meta-analysis of prospective studies. Journal of the American Medical Association. 1998; 279(15): 1200-5.

6. Anderson JG, Jay SJ, Anderson M, Hunt TJ. Evaluating the capability of information technology to prevent adverse drug events: A computer simulation approach. Journal of the American Medical Informatics Association. 2002; 9(5): 479-90.

7. Bates DW, Cullen DJ, Laird N, Petersen LA, Small SD, Servi D, Laffel G, Sweitzer BJ, Shea BF, Hallisey R, et a. Incidence of adverse drug events and potential adverse drug events. Implications for prevention. ADE prevention study group. Journal of the American Medical Association. 1995; 274(1): 29-34.

8. Leape LL, Bates DW, Cullen DJ, Cooper J, Demonaco HJ, Gallivan T, Hallisey R, Ives J, Laird N, Laffel G, et a. Systems analysis of adverse drug events. ADE prevention study group. Journal of the American Medical Association. 1995; 274(1): 35-43.

9. Vincent C, Neale G, Woloshynowych M. Adverse events in British hospitals: Preliminary retrospective record review. BMJ (British Medical Journal). 2001; 322(7285): 517-9.

10. Bell DS, Cretin S, Marken RS, Landman AB. A conceptual framework for evaluating outpatient electronic prescribing systems based on their functional capabilities. Journal of the American Medical Informatics Association. 2004; 11(1): 60-70.

11. Burnum JF. Preventability of adverse drug reactions [letter]. Annals of Internal Medicine. 1976; 85:80(1).

12. Gurwitz JH, Field TS, Harrold LR, Rothschild J, Debellis K, Seger AC, Cadoret C, Fish LS, Garber L, Kelleher M, Bates DW. Incidence and preventability of adverse drug events among older persons in the ambulatory setting. Journal of the American Medical Association. 2003; 289(9): 1107-16.

13. Nebeker JR, Hoffman JM, Weir CR, Bennett CL, Hurdle JF. High rates of adverse drug events in a highly computerized hospital. Archives of Internal Medicine. 2005; 165(10): 1111-6.

14. Kaushal R, Bates DW, Landrigan C, McKenna KJ, Clapp MD, Federico F, Goldmann DA. Medication errors and adverse drug events in pediatric inpatients. Journal of the American Medical Association. 2001; 285(16): 2114-20.

15. van den Bemt PM, Fijn F, van der Voort PH, Gossen AA, Egberts TC, Brouwers JR. Frequency and determinants of drug administration errors in the intensive care unit. **Critical Care Medicine**. 2002; 30: 846-50.

16. Tang F-I, Sheu S-J, Yu S, Wei I-L, Chen C-H. Nurses relate the contributing factors involved in medication errors. Journal of Clinical Nursing. 2007; 16: 447-57.

17. Bobb A, Gleason K, Husch M, Feinglass J, Yarnold PR, Noskin GA. The epidemiology of prescribing errors: The potential impact of computerized prescriber order entry. Archives of Internal Medicine. 2004; 164(7): 785-92.

18. Bates DW, Kuperman GJ, Wang S, Gandhi T, Kittler A, Volk L, Spurr C, Khorasani R, Tanasijevic M, Middleton B. Ten commandments for effective clinical decision support: Making the practice of evidence-based medicine a reality. Journal of the American Medical Informatics Association. 2003; 10(6): 523-30.

19. Gross PA, Bates DW. A pragmatic approach to implementing best practices for clinical decision support systems in computerized provider order entry systems. Journal of the American Medical Informatics Association. 2006: M2173.

20. Bobb AM, Payne TH, Gross PA. Viewpoint: Controversies surrounding use of order sets for clinical decision support in computerized provider order entry. Journal of the American Medical Informatics Association. 2007; 14(1): 41-7.

21. Ray MN, Houston TK, Yu FB, Menachemi N, Maisiak RS, Allison JJ, Berner ES. Development and testing of a scale to assess physician attitudes about handheld computers with decision support. Journal of the American Medical Informatics Association. 2006; 13(5): 567-72.

22. Nelson NC, Evans RS, Samore MH, Gardner RM. Detection and prevention of medication errors using real-time bedside nurse charting. Journal of the American Medical Informatics Association. 2005; 12(4): 390-7.

23. Villanueva M. Nursing calculators. http://www.manuelsweb.com/nrs_calculators.htm.

24. Medical Wizards. http://www.medicalwizards.com

25. Comet. This involves a potential partnership between *Comet* (http://www.ausindustry.gov.au) and *Grey Innovations* (http://www.greyinnovation.com).

26. Kuperman GJ, Bobb AM, Payne T, Avery AJ, Gandhi TK, Burns G, Classen

D, Bates DW. Medication-related clinical decision support in computerized provider order entry systems: A review. Journal of the American Medical Informatics Association. 2006: M2170.

27. Rosenbloom ST, Geissbuhler AJ, Dupont WD, Giuse DA, Talbert DA, Tierney WM, Plummer WD, Stead WW, Miller RA. Effect of cpoe user interface design on user-initiated access to educational and patient information during clinical care. Journal of the American Medical Informatics Association. 2005; 12(4): 458-73.

28. Johnson CW. Why did that happen? Exploring the proliferation of barely usable software in healthcare systems. Quality & Safety in Health Care. 2006; 15(suppl_1): i76-81.

29. Miller RA, Gardner RM. Recommendations for responsible monitoring and regulation of clinical software systems. Journal of the American Medical Informatics Association. 1997; 4(6): 442-57.

30. 3Miller RA, Gardner RM, Johnson KB, Hripcsak G. Clinical decision support and electronic prescribing systems: A time for responsible thought and action. Journal of the American Medical Informatics Association. 2005; 12(4): 403-9.

31. Federal Register (US). July 14, 1996. 36886-7.

32. Murff HJ, Kannry J. Physician satisfaction with two order entry systems. Journal of the American Medical Informatics Association. 2001; 8(5): 499-511.

33. Tamblyn R, Huang A, Kawasumi Y, Bartlett G, Grad R, Jacques A, Dawes M, Abrahamowicz M, Perreault R, Taylor L, Winslade N, Poissant L, Pinsonneault A. The development and evaluation of an integrated electronic prescribing and drug management system for primary care. Journal of the American Medical Informatics Association. 2005: M1887.

34. Council on Clinical Information Technology. Electronic prescribing systems in pediatrics: The rationale and functionality requirements. Pediatrics. 2007; 119(6): 1229-31.

35. Lesar TS, Briceland L, Stein DS. Factors related to errors in medication prescribing. Journal of the American Medical Association. 1997; 277(4): 312-7.

36. Herout PM, Herstad BL. Medication errors involving continuously infused medications in a surgical intensive care unit. Critical Care Medicine. 2004; 32: 428-32.

37. Schrier RW, Gambertoglio JG, Aweeka FT, Schrier DM, Austin JL, Heiland CD, McNamara TJ. Providing patient-specific drug information. 1998; US Patent 5833599.

38. Monson RC. Administration of drugs in Publication Number. 2007; World Intellectual Property Organization WO/2007/019606.

39. Rosenbloom ST, Miller RA, Johnson KB, Elkin PL, Brown SH. Interface terminologies: Facilitating direct entry of clinical data into electronic health record systems. Journal of the American Medical Informatics Association. 2006; 13(3): 277-88.

40. Kuperman GJ, Gibson RF. Computer physician order entry: Benefits, costs, and issues. Annals of Internal Medicine. 2003; 139(1): 31-9.

41. Payne TH, Hoey PJ, Nichol P, Lovis C. Preparation and use of preconstructed orders, order sets, and order menus in a computerized provider order entry system. Journal of the American Medical Informatics Association. 2003; 10(4): 322-9.

42. Rothschild AS, Lehmann HP. Information retrieval performance of probabilistically generated, problem-specific computerized provider order entry picklists: A pilot study. Journal of the American Medical Informatics Association. 2005; 12(3): 322-30.

43. Tang SS, Diamond C, Arouh S. Neural network drug dosage estimation. 2003; US Patent 6658396.

44. Reichley RM, Seaton TL, Resetar E, Micek ST, Scott KL, Fraser VJ, Dunagan WC, Bailey TC. Implementing a commercial rule base as a medication order safety net. Journal of the American Medical Informatics Association. 2005; 12(4): 383-9.

45. Brennan PF. Cpoe: Sufficient, but not perfect, evidence for taking action. Journal of the American Medical Informatics Association. 2007; 14(1): 130-1.

46. Osheroff JA, Teich JM, Middleton B, Steen EB, Wright A, Detmer DE. A roadmap for national action on clinical decision support. Journal of the American Medical Informatics Association. 2007; 14(2): 141-5.

47. Boxwala AA, Dierks M, Keenan M, Jackson S, Hanscom R, Bates DW, Sato L. Organization and representation of patient safety data: Current status and issues around generalizability and scalability. Journal of the American Medical Informatics Association. 2004; 11(6): 468-78.

48. Frankel A, Gandhi TK, Bates DW. Improving patient safety across a large integrated health care delivery system. *International Journal for Quality in Health Care*. 2003; 15(suppl_1): i31-40.

49. Bates DW. Using information technology to reduce rates of medication errors in hospitals. BMJ (British Medical Journal). 2000; 320(7237): 788-91.

50. Dastur KM, Koenig PA, Wershmidt G, Hong S, Jiang J. Automated infusion system with dose rate calculator. 1988; US Patent 57772635.

51. Farrelly PA. Hand held apparatus for performing medical calculations. 1997; US Patent 5630664.

52. Bishko JR, Single TG, Stone D. Electronic fluid flow rate controller for controlling the infusion of intravenous drugs into a patient. 1992; US Patent 5104374.

53. Ramsay OB, Heezen A. Tutorial device and method for determining drug dosages. 1999; US Patent 5915971.

54. Simons TD. Handheld medical calculator and medical reference device. 2000; US Patent 6167412.

55. Francis KR, Mathis DM, Conkright G. Intensive care calculator : Medication dose calculator. 2002; US Patent 20060129357.

56. Izumi M, Iguchi S, Inoue F. Drip blender, mixing tube, liquid drug container, liquid mixture container, drip blending system and method of blending drip. 2004; World Intellectual Property Organization WO/2004/045489.

57. Fritzson P, Baltzer L. Computer system for determining a drug dosage. 2005; World Intellectual Property Organization WO/2005/041105.

Correspondence

Dr. Ron Monson
Student Learning Unit
Victoria University
Melbourne Vic 3011, Australia
MobilePhone:0411267348
ronald.monson@vu.edu.au

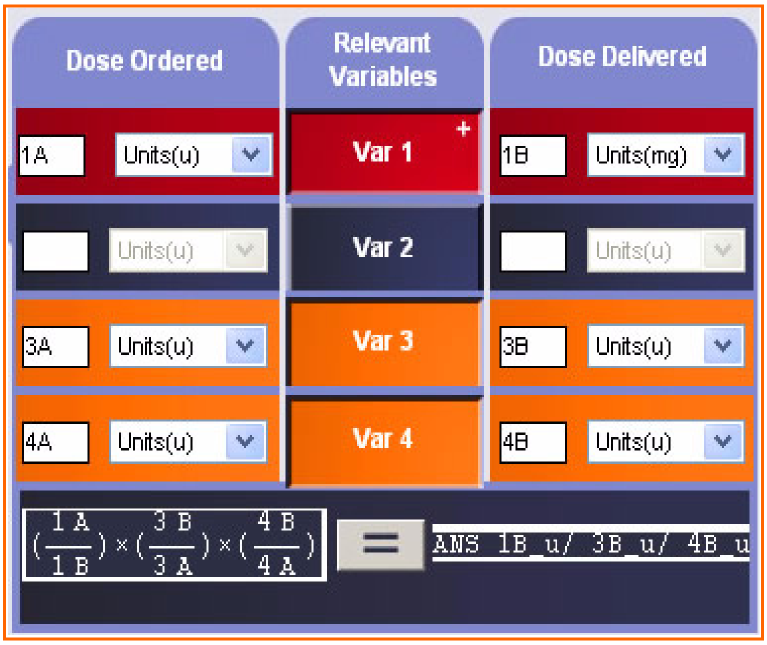
5. Appendix

In this appendix we describe the operation of the Drugulator interface and how inputs are used to construct a mathematical expression that is evaluated on submission to the computational back-end. By way of several examples we then show how this operation implements a unifying principle that allows a wide array of dosage calculations to be similarly automated. Finally, a description of the technical infrastructure and accompanying website is described.

5.1. Forming the Mathematical Expression

We first show the mechanics of how values input into the Drugulator's input fields are combined into an arithmetical expression before later showing how this expression corresponds to the desired dosage. This forming of the appropriate mathematical expression corresponds to the "Formulaic-Aided Dosing" used as part of our original FCAD definition. In Figure 4, the (red or orange) *activated* rows contain inputs that are used to generate a quotient that is used in the dosage calculation.

If a row is activated as *red* then the quotient is formed by taking the L.H.S. value of the row (in the *Dose Ordered* Column) and dividing by the R.H.S. value of the row (in the *Dose Delivered* Column). In the example of Figure 4, the quotient so generated is $\frac{1A}{1B}$. Conversely, if a row is activated as *orange*, then the quotient formed reverses the roles of denominator and numerator. That is, the quotient is formed by taking the R.H.S. value (appearing in the *Dose Delivered* Column) and dividing by the L.H.S. value (appearing in the *Dose Ordered* Column). From Figure 4 the quotients so generated are $\frac{3B}{3A}$ and $\frac{4B}{4A}$.



Dose Ordered		Relevant Variables	Dose Delivered	
1A	Units(u)	Var 1 +	1B	Units(mg)
	Units(u)	Var 2		Units(u)
3A	Units(u)	Var 3	3B	Units(u)
4A	Units(u)	Var 4	4B	Units(u)

$(\frac{1A}{1B})\times(\frac{3B}{3A})\times(\frac{4B}{4A})$ = ANS 1B_u/ 3B_u/ 4B_u

Figure 4: Demonstrating how the interface's various fields automatically define a mathematical expression that is to correspond to a correct drug dosage.

All the quotients so formed from the activated rows are then multiplied together to form a product (as shown on the left of the "=" button). This product is then evaluated to a numerical answer and placed to the right of the "=" button once it is clicked. This evaluated answer is typically the amount of the dose to be administered. Apart from some usage details to be described later (relating to the management of units and activation of rows), this essentially describes the mechanics of this implementation of the Drugulator method.

5.2. Implementing Proportional Relationships

As will be demonstrated, these mechanics implement *proportional* relationships that typically exist between the final dosage and those other factors that affect its final value. Mathematically, these are proportional relationship involving a single *dependent* variable and other *independent* variables. In the Drugulator method, individual variables are assigned fixed rows in the interface and their dependent/ independent status is then activated by clicking their corresponding middle buttons within the *Relevant Variables* column. The resulting status is then indicated by each rows' background colour under the following identification: Dependent ↔ Red; Independent ↔ Orange; Inert ↔ Blue[1].

Hence we see that in the example of Figure 4, Var 1 has been selected to be the *dependent* variable while Var 3 and Var 4 have been selected as the *independent* variables. Var 2 has been left inert and therefore plays no part in this particular calculation. Having identified the dependent variable (typically the dose amount) and the independent variables (those variables affecting the dependent variable) it remains to mathematically illustrate the relationship between these two types of variables. As mentioned previously, it turns out that for the vast majority of dosage calculations required at the administration stage, the independent variables affect the dosage (dependent) variable *proportionally*. That is, changing an independent variable by a certain factor means that the base dose also needs to change by this same factor. This simple principle, underpins a surprisingly wide array[2] of dosage calculations and the way in which the interface's mechanics implements this principle can be observed as follows:

Dependent Variable: In the arithmetical expression of Figure 4, the factor contributed by the red row (dependent variable Var 1) is $\frac{1A}{1B}$.

The term *1A* can be interpreted as the base dose with *1B* the units (sometimes defined per tablet) in which this base is to be delivered. The quotient $\frac{1A}{1B}$ therefore effectively ensures the final dosage amount is in the correct units.

Independent Variables: It is through *changes* in the independent variable(s) that their proportionate relationship to the dependent variable is implemented. These changes are from the values specified in the *Dose Ordered* column to those value specified in the *Dose Delivered* column so that the factor change is simply the latter divided by the former. This can be seen algebraically from the contribution of the orange rows in Figure 4.

Ans $= \frac{1A}{1B} \times (3F) \times (4F)$ with $(3F) = \frac{3B}{3A}$, $(4F) = \frac{4B}{4A}$.

Since $3A \times 3F = 3B$ the term $3F$ represents the factor by which $3A$ changes in becoming $3B$. Equivalently, $3F$ represents the factor by which the row's corresponding independent variable has changed from its value used in the base dose. Therefore, due to the assumed proportional relationship between the dependent and independent variable(s), the final delivered dose needs to also change by a factor of $3F$. That is, the delivered dose - $(\frac{1A}{1B})$ - needs to be adjusted by multiplying it by this factor $3F$. A similar argument applies for $4F$ (and indeed any number of other independent variables should they be subsequently defined) with the collective effect of these factor changes being captured in the final product - $\frac{1A}{1B} \times (3F) \times 4F$

This completes the initial description of the Drugulator's essential action save for one minor point regarding the specification of *units*. In Figure 4 the selected fields all contained the same values. This simplified the description of the mechanics behind the interface's formulae construction but clearly in practice the possibility of variable unit specifications needs to be incorporated. In these cases the terms $1A$, $1B$, $1C$ are suitably modified (with conversion factors between different selected units) as is described later in the section *Specifying Units*. Before describing some of the other features we give three basic examples.

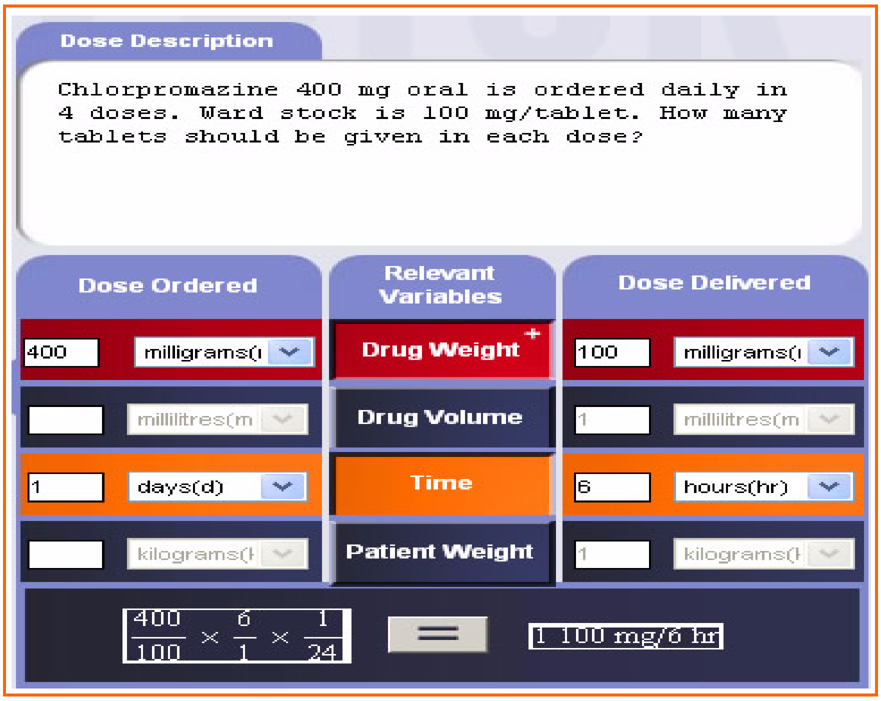
Example 1

A dose is to be calculated using the following information which appears in the *Dose Description* panel of the interface as shown in Figure 5.

Chlorpromazine 400 mg oral is ordered daily in 4 doses. Ward stock is 100 mg/tablet. How many tablets should be given in each dose?

[1] Note that the dependent variable also has a non-color identification via a small white cross appearing in the top right-hand corner.

[2] Surprising in the sense that traditional nursing education usually develops individual formulae for each dosage calculation thereby obscuring this single proportionality principle.



Dose Description

Chlorpromazine 400 mg oral is ordered daily in 4 doses. Ward stock is 100 mg/tablet. How many tablets should be given in each dose?

Dose Ordered		Relevant Variables	Dose Delivered	
400	milligrams(	Drug Weight +	100	milligrams(
	millilitres(m	Drug Volume	1	millilitres(m
1	days(d)	Time	6	hours(hr)
	kilograms(	Patient Weight	1	kilograms(

$$\frac{400}{100} \times \frac{6}{1} \times \frac{1}{24}$$ = 1 100 mg/6 hr

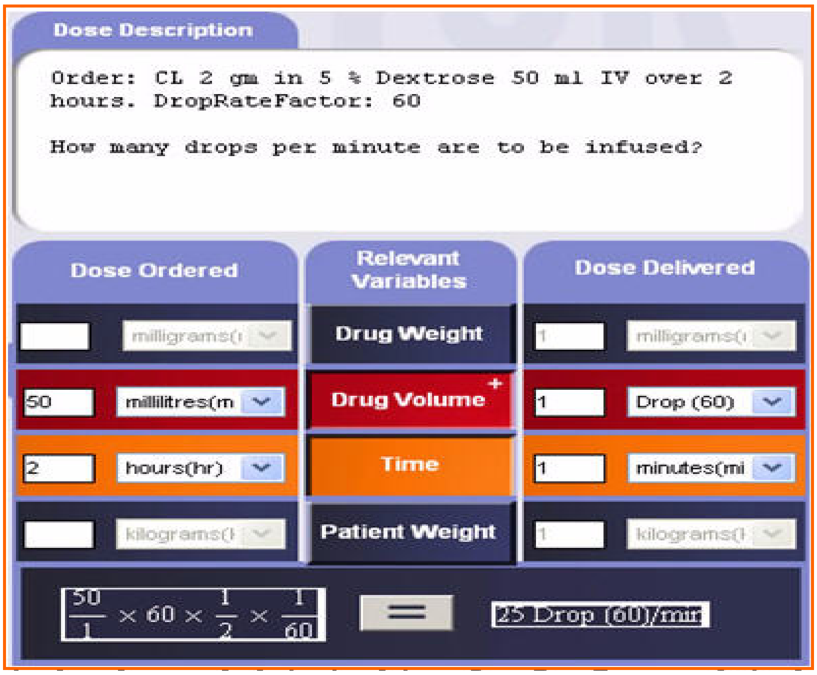
Figure 5: A simple example involving a Tablet Calculation

The question requests a number of tablets which corresponds to a *Drug Weight* - hence this is the main dependent variable as reflected by this row's red colour in Figure 5. The amount of this *Drug Weight* in any single dose depends on the frequency of administration; that is, it depends on the independent *Time* variable as reflected in that row's orange colour. The base dose is 400 mg per (1) day as entered in the *Dose Ordered* panel. This base dose needs to be varied firstly in relation to the *Drug Weight* where 100 mg units are effectively being defined given that this is the mass of each tablet. The base dose is being delivered in 4 doses which over 1 day equates to a dose every 6 hours. These variations are both specified in the *Dose Delivered* panel. Finally, note that the generated answer (of 1 tablet) includes the units with which it is to be administered (i.e. with 100mg units and every 6 hours).

Example 2

Order: CL 2 gm in 5 % Dextrose 50 ml IV over 2 hours. DropRateFactor: 60

How many drops per minute are to be infused?



Order: CL 2 gm in 5 % Dextrose 50 ml IV over 2 hours. DropRateFactor: 60

How many drops per minute are to be infused?

Dose Ordered		Relevant Variables	Dose Delivered	
	milligrams(!	Drug Weight	1	milligrams(!
50	millilitres(m	Drug Volume +	1	Drop (60)
2	hours(hr)	Time	1	minutes(mi
	kilograms(!	Patient Weight	1	kilograms(!

$\frac{50}{1} \times 60 \times \frac{1}{2} \times \frac{1}{60}$ = 25 Drop (60)/min

Figure 6: Showing how dosage calculation involving a *Drop Rate Factor* can be implemented in the Drugulator's interface.

This example (transcribed into the interface as shown in Figure 6) involves a standard "DropRateFactor" calculation usually performed using a learned formula. This formula however, also embodies a proportionate relationship between dose and independent variables and hence can be calculated within the structure of the Drugulator interface.

The request "how many drops per minute" indicates that a *Drug Volume* (in drops) is the sought-after dependent variable with the number of drops depending on the time over which the drops are administered; that is, with the amount depending on the independent *Time* variable. This is reflected in the *Drug Volume* and *Time* variables being specified as the respective independent (red) and dependent (orange) variables.

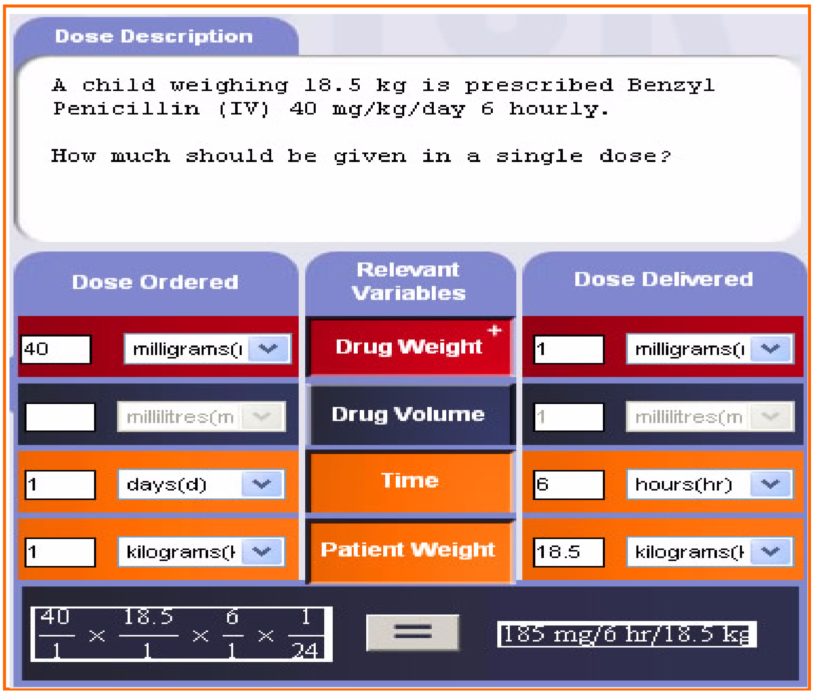
The base dose of 50 ml/2 hrs is then entered in the respective positions of the *Dose Ordered* panel. This base dose is being varied only in the units in which it is to be delivered. The standard terminology of a "Drop Rate Factor" of 60 indicates that the dose is to be delivered in drops of volume $\frac{1}{60}$ ml each. This is reflected in the *Drop* (60) setting in the *Dose Delivered* Panel with the *Drop (60)* term therefore representing a volume unit of $\frac{1}{60}$ ml[1]. Finally, the time unit is to be changed to minutes as entered in the *Dose Delivered* Panel. Again note that the final answer of 25 is followed by the units to which the base dose has been changed.

Example 3

The following example was first described in the *System Description* section and its solution in the Drugulator interface is repeated below in Figure 7.

A child weighing 18.5 kg is prescribed Benzyl Penicillin (IV) 40 mg/kg/day 6 hourly. How much should be given in a single dose?

[1] Note that placing 1/60 in the input field and choosing ml as the units would also yield the correct answer.



Dose Description

A child weighing 18.5 kg is prescribed Benzyl Penicillin (IV) 40 mg/kg/day 6 hourly.

How much should be given in a single dose?

Dose Ordered		Relevant Variables	Dose Delivered	
40	milligrams(	Drug Weight +	1	milligrams(
	millilitres(m	Drug Volume	1	millilitres(m
1	days(d)	Time	6	hours(hr)
1	kilograms(k	Patient Weight	18.5	kilograms(k

$\frac{40}{1} \times \frac{18.5}{1} \times \frac{6}{1} \times \frac{1}{24}$ = 185 mg/6 hr/18.5 kg

Figure 7: Showing a paediatric weight-based dosage calculation.

Based on the previous discussion, the identification of *Drug Weight* as the dependent variable and *Time* and *Patient Weight* as the independent variables follows naturally from the description while the proportional relationship between them, as before, ensures that the constructed mathematical expression corresponds to the required dose. Note that although not explicitly stated, it is assumed that the Drug Weight is to be delivered in the same units as that provided in the dose (mg) and that, as in the previous two examples, the final independent units follow the calculated answer of 185 mg.

5.3. Specifying Dependent/ Independent Variables

One of the reasons behind the Drugulator's flexibility (and indeed of all so-called *Constructors* as defined in the *Discussion* section) is that for different scenarios, different variables can be identified as dependent, independent or inert. In the Drugulator interface, each row is identified with a variable via a button label contained in the middle *Relevant Variables* panel. A variable is then identified as being dependent/independent (or irrelevant) depending on whether or not its button label was the *first* one clicked. That is; the first variable button label clicked identifies its variable as the *dependent* variable while subsequent clicks on any of the remaining button labels identify the corresponding variables as *independent*. As mentioned earlier, these identifications are reflected in a colour change of each variable's associated row with red as dependent and orange as independent.

So, for example, the identification of Figure 7 can be made by simply clicking the buttons in the following order - *Drug Weight*, *Time*, *Patient Weight*. (N.B. the order *Drug Weight*, *Patient Weight*, *Time* also makes the same identification). Note that removing a variable's identification is performed by clicking again on its button label and is indicated by having the variable's row return to its "inert" blue colour[1]. Note also that this variable identification is a necessary first step since activating a variable's row makes its row's inputs editable (and is indicated by the "undimming" of these corresponding inputs).

It is easy to see that the aforementioned algorithm for selecting dependent/independent variables always produces exactly one dependent variable and (from the remaining variables) between 0-3 independent variables. There are constraints however, associated with dosage calculations that make certain configurations more likely than others. For example, given that a drug's weight or volume is most often the primary quantity of interest; the dependent variable is typically either *Drug Weight* or *Drug Volume*. Note also the inbuilt extensibility whereby other variables can potentially be defined using additional rows and/or other relationships *other* than the current proportional one between dependent and independent variables. This was discussed earlier in

[1] If a dependent variable is rendered inert then the next inert variable label clicked identifies it as the new dependent variable. If an independent variable is rendered inert then subsequent variable clicks identify the respective variables as the new, independent variables.

describing the relevance of a *Constructor's* idiom to CPOE systems.

5.3.1. Specifying Units

Converting between different units is not really part of the conceptual understanding involved in calculating standard drug dosages but is instead, merely a reflection of the disparity between the drug *packaging* and drug *prescription* quantities. Fortunately, in the Drugulator this task no longer interferes with a dose's mathematical conception since all conversions are handled automatically. Recall from Figure 4 how all the input values of $1A$, $1B$, $1C$ ... etc are measured in the *same* units - namely *Units (u)*. Making such an assumption allowed the essence of a dose's construction from the interface's input fields to be more clearly observed. Given that each row is identified with a particular variable however, the actual range of units selectable in that row will clearly depend on the units with which that variable is typically measured. So for example, the units in the *Drug Weight*[1] row contains a drop-down menu consisting of "micrograms (mcg)"[2], "milligrams (mg)", "grams (g)", "kilograms (kg)" which covers the range of units needed to describe this variable[3]. The variable *Patient Weight*, on the hand, only contains the units "grams" and "kilograms" since these are invariably sufficient to describe a person's body weight.

When different units are specified in any one row, a conversion factor is required, effectively to ensure that both the denominator and numerator are being measured using the same scale in the corresponding quotient. This factor effectively converts the units defined in the *Dose Ordered* column into those units defined in the *Dose Delivered* column. When the variable is the dependent variable, this factor appears as the numerator (because of the way the quotient is formed): when the variable is an independent variable the factor appears on the denominator (again because of the reciprocal quotient). For example, in the calculation of Fig 4, *Drug Volume* is the dependent variable with respective units, "ml" and "Drop 60". As changing from ml to Drop 60 units ($\frac{1}{60}$ th of a ml) involves a factor of 60, this appears on the numerator. *Time* is the independent variable in this example with respective units, hours and minutes. As changing from hours to minutes involves a factor of 60, this factor appears on the denominator.

Any answer generated by the Drugulator is always followed by unit abbreviations corresponding to the unit specifications of the independent variables in the *Dose Delivered* panel. These can be viewed as a reminder of the new values taken by those independent variables that affect the patient's calculated dose. Following normal convention, The Drugulator separates these abbreviations with the symbol "/" (read as "per") as well as dropping any unity specifications. For example, 8 mcg/min/kg conventionally denotes: "8 micrograms per "1" minute and per "1" kilogram" (although note that the omitted 1's still needs to be included in the *Drug Delivered* panel).

Less conventionally, but we argue more logically and safely, The Drugulator *never* omits or alters any abbreviated units that have been used in the calculation. In particular, it always includes the relevant units specified in the *Drug Delivered* panel. Conventionally these values are sometimes omitted being assumed from the context although we believe this is a potential ambiguity that can lead to unnecessary confusion about what independent variables have been incorporated. Hence, in the three previous examples, the respective answers of: 1 Tablet, 25 Drops and 185mg are traditionally sufficient whereas the respective Drugulator outputs of: 1 100mg/6hr, 25 Drop (60)/min and 185 mg/6 hr/18.5 kg reinforces that: 1 100mg tablet needs to be administered every 6 hours; that 25 1/60 ml drops need to be infused every minute and finally, that the 185 mg dose needs to occur for a 18.5 kg patient every 6 hours.

5.3.2. Stock Concentration Interface

Drugs are often stored as mixtures from which drug weight or drug volume amounts are measured out depending on the prescription type. Furthermore, within the mixture, the drug weight and drug volume are *proportionally* related in a way usually defined as the mixture's *concentration*. But both *Drug Weight* and *Drug Volume* are variables of the Drugulator's interface which recall, also implements proportional relationships; hence it is ready made to perform any preliminary concentration calculations. Such usage also dovetails nicely with the Drugulator's idiom of reducing the number of possible inputs by re-using arguments. On the other hand, outputs from stock concentration calculations are invariably "plugged

[1] Strictly speaking these are units of *Mass* so that it would actually be more accurate to name the variable *Drug Mass* instead of *Drug Weight* (which is a Force). The latter has been chosen to follow standard clinical terminology in Australia.

[2] Note that strictly speaking, the SI abbreviation for "microgram: is " µ" but to avoid potential confusion with "g" it is standard to instead use "mcg"

[3] In principle making the unit specifications "inputable" (as opposed to the current "selectable") would extend the range of possible units. This may be useful if one would one to use some other SI abbreviation (pico, nano, etc) or other unit systems (e.g. Imperial) although it would also probably unnecessarily complicate the interface for the majority of calculations. (Future versions may at least provide this as an option).

into" the Drugulator interface meaning that it would then need to be activated twice for any dosage requiring a preliminary concentration calculation. This was a source of some confusion (perhaps also because this repeated usage renders the panels headings with a double meaning) in an early piloted version so it was decided to create a separate interface dedicated to these preliminary stock-concentration calculations. This additional interface adds clarity in terms of sequencing the concentration calculations while the complexity introduced by adding new inputs is minimized given that the same idiom of dependent/ independent variables transfers from the original interface. We now illustrate a stock-concentration calculation.

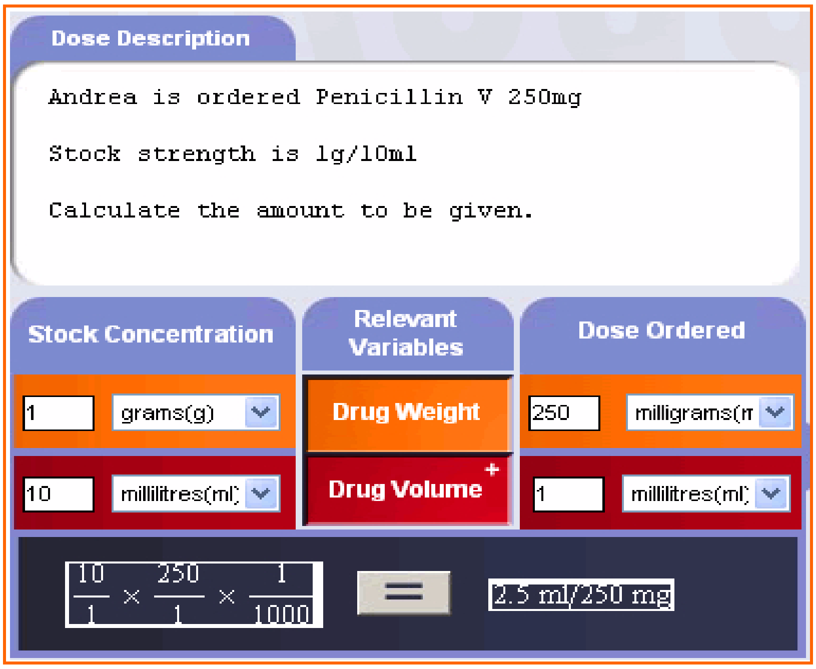
Example 4

The following example amounts to finding an equivalent drug volume of a drug weight based on a given stock concentration (see Figure 8).

Andrea is ordered Penicillin V 250mg

Stock strength is 1g/10ml

Calculate the amount to be given.



Dose Description

Andrea is ordered Penicillin V 250mg

Stock strength is 1g/10ml

Calculate the amount to be given.

Stock Concentration		Relevant Variables	Dose Ordered	
1	grams(g)	Drug Weight	250	milligrams(n
10	millilitres(ml)	Drug Volume	1	millilitres(ml)

$\frac{10}{1} \times \frac{250}{1} \times \frac{1}{1000}$ = 2.5 ml/250 mg

Figure 8: Showing how a preliminary Stock Concentration calculation can be performed in the Drugulator's L.H.S. sub-interface.

Note that while this particular calculation can be straightforwardly performed by inspection, more cumbersome values (and unit conversions) can still render such an interface a useful calculating tool. The real point however, is such a calculation also embodies a proportional relationship between relevant variables and hence also can be performed within the Drugulator interface. In particular, this push for uniform automation within a single interface (to obtain all the potential advantages previously discussed) can be extended to include stock-concentration calculations. Next we clarify the nature of this extension when applied to these stock-concentration calculations.

Recall that a dependent variable - the red row- is being identified as the main quantity of interest and is set to depend *in proportion* to an independent variable in the orange row. As can be seen in Figure 8 this *in proportion* dependence is set up in stock concentration calculations in the same way as described earlier for the R.H.S. interface. That is, if the row is red the quotient formed is the L.H.S. value divided by the R.H.S. value ($\frac{10}{1}$ in Figure 8): if the row is orange, the quotient formed is the R.H.S. value divided by the L.H.S. value ($\frac{250}{1}$). As before, unit changes across any row are accounted for using an appropriate conversion factor ($\frac{1}{1000}$). Finally, note that the generated output continues to include the final independent variable amount (250 mg).

There are two modifications specific to this stock-concentration interface. Firstly, since in any stock concentration calculation, it is always the case that there must exist exactly one dependent variable and one independent variable (i.e. exactly one red row and one orange row) any initial click defining the corresponding row as the dependent variable (thereby making it red) automatically causes the other variable to become independent (thereby making it orange). For example, the dependent/independent identification of Figure 8 arises from the *single* click of the *Drug Volume* button label.[1]

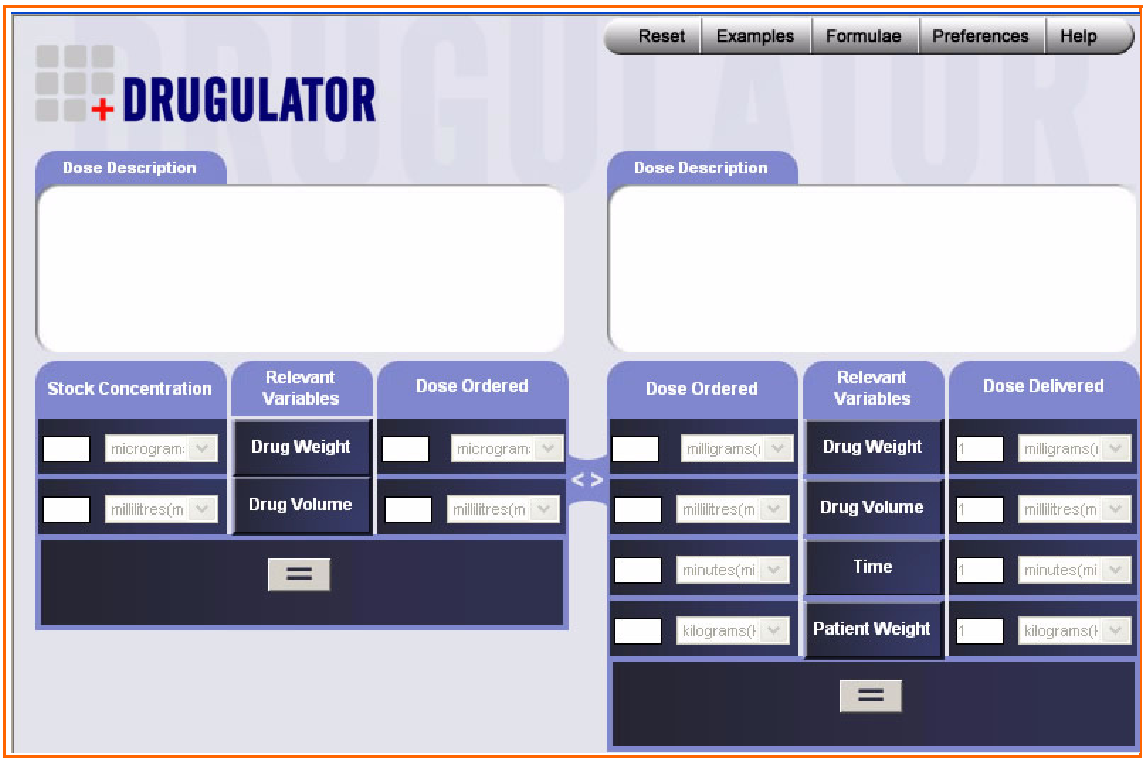


Figure 9: The Drugulator Interface before starting a Drug Calculation.

The other modification in this interface involves the re-labelling of the left-hand side (L.H.S.) and right-hand side (R.H.S.) panels to respectively *Stock Concentration* and *Dose Ordered*. The rationale behind such labelling is that mathematically, the L.H.S. holds the base values of any selected variables whereas the R.H.S. represents how these values are changed. That is, the Stock Concentration's panel contains *Drug Weight* and *Drug Volume* values which represent a base setting to be altered according to the *Dose Ordered* amount. Note the same *Dose Ordered* labelling of the inner, adjacent panels of both interfaces. This reflects the sequence whereby any *Drug Volume* (*Drug Weight*) value generated by the stock concentration interface is typically "plugged" back into *Drug Volume* (*Drug Weight*) input fields of the R.H.S. interface in the *Dose Ordered* panel[2]. An example of this sequence is contained in the next example.

A Bi-interface Example.

The final example shows the two interfaces working together and represents the more sophisticated type of calculation possible in the Drugulator system. Recall the initial (inert) Drugulator interface showing the left-hand "concentration calculating" interface adjacent to the right-hand side "dosage calculating" interface (Figure 9).

Example 5

Order: Dopamine infusion 5 mcg/kg/min for a 90 kg patient.

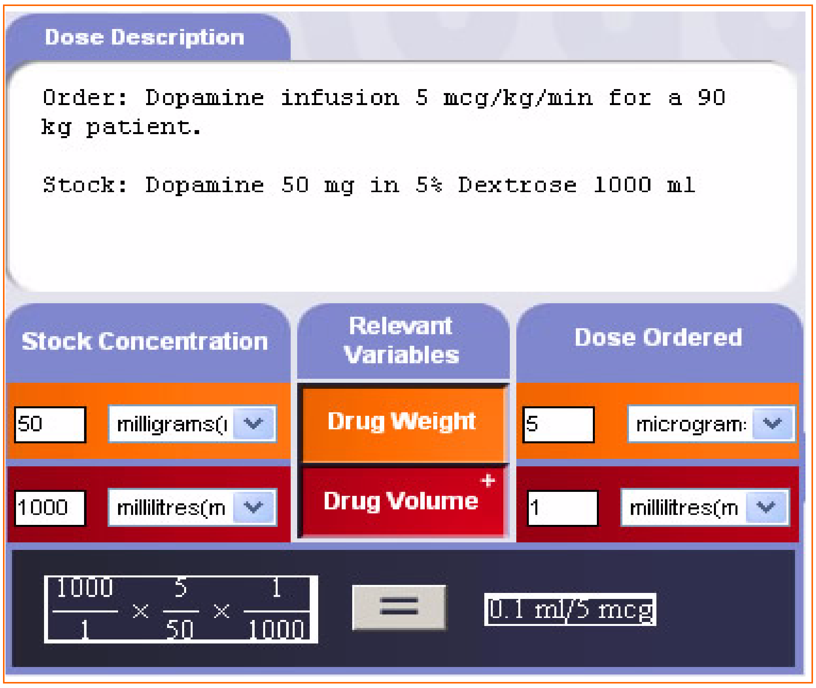
Stock: Dopamine 50 mg in 5% Dextrose 1000 ml

How many ml/hr are to be infused?

First a preliminary calculation is required to find a volume equivalent to 5 mcg using the given stock concentration. This calculation is performed in the L.H.S. interface as depicted in Figure 10.

[1] This is the most common situation but the other alternative where a Drug Weight is required based on a quoted Drug Volume (e.g. when checking the output of any previous stock concentration calculation) can also be performed with a single click of the Drug Weight button.

[2] It is natural to want to automate this "plugging in" process. The "connecting valve" between the two interfaces (shown as "<>") has been earmarked to perform and indicate such an extension.



Order: Dopamine infusion 5 mcg/kg/min for a 90 kg patient.

Stock: Dopamine 50 mg in 5% Dextrose 1000 ml

50	milligrams(ı	Drug Weight	5	microgram:
1000	millilitres(m	Drug Volume	1	millilitres(m

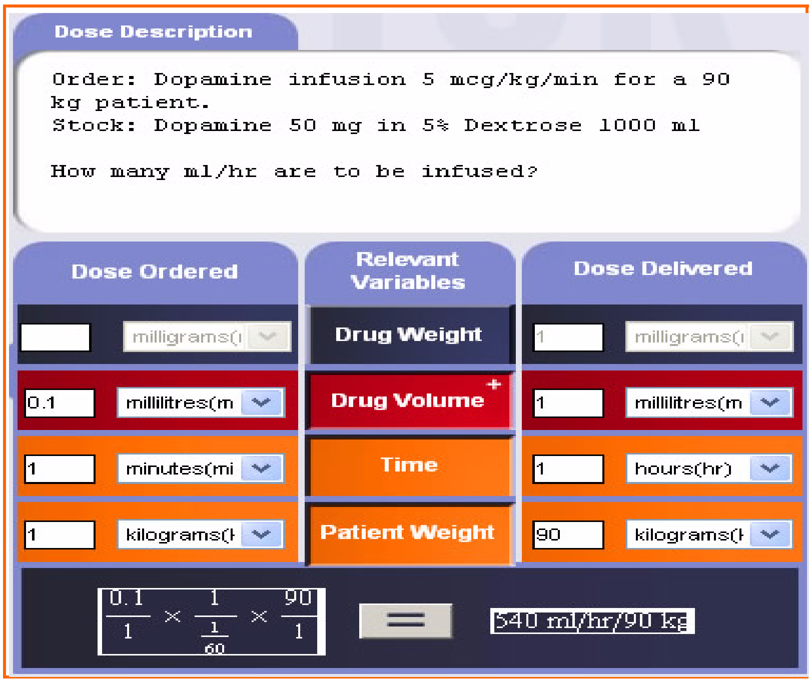
$$\frac{1000}{1} \times \frac{5}{50} \times \frac{1}{1000} = 0.1 \text{ ml}/5 \text{ mcg}$$

Figure 10: Showing the preliminary stock concentration calculation in a more involved example.

This stock concentration calculation shows that using the supplied stock, 5 mcg is equivalent to 0.1 ml and hence a dose of 5 mcg/kg/min is equivalent to 0.1 ml/kg/min as entered in the *Dose Ordered* panel of the R.H.S. interface as shown in Figure 11.

The variation on this base dose for this particular administration involves a 90 kg patient and a dose frequency *per hour* as specified in the *Dose Delivered* panel of Figure 11. Finally, on clicking the equal sign the final dose of 540 ml/hr/90kg is generated.

Even in this more involved example, the calculation's individual steps are still mathematically trivial given that they all ultimately reduce to fraction multiplication. When contextualized within the demands of a clinical situation however, the necessity for such computer-aided assistance becomes more apparent (apart from persistently high ADE rates). For example, consider the accompanying tasks required of health-care workers: they need to; apply natural language interpretations to form the correct fraction, evaluate the fraction either by hand or calculator (while remaining alert to the possibilities of required unit conversions), perform any final checks, deal with the physicality of dose measures and administration all under considerable time pressure. Further, these demands become compounded with increases in the complexity of dosage regimes (that incorporate, for example, additional variables) to the point that, we argue point-of-care calculation becomes infeasible *without* this type of FCAD assistance. The structure provided by the Drugulator provides a consistent framework in which the natural language interpretation, mathematical evaluation and final checks can be automated.



Dose Description

```
Order: Dopamine infusion 5 mcg/kg/min for a 90
kg patient.
Stock: Dopamine 50 mg in 5% Dextrose 1000 ml

How many ml/hr are to be infused?
```

Dose Ordered		Relevant Variables	Dose Delivered	
	milligrams(ı	Drug Weight	1	milligrams(ı
0.1	millilitres(m	Drug Volume	1	millilitres(m
1	minutes(mi	Time	1	hours(hr)
1	kilograms(k	Patient Weight	90	kilograms(k

$$\frac{0.1}{1} \times \frac{1}{\frac{1}{60}} \times \frac{90}{1}$$

=

540 ml/hr/90 kg

Figure 11: Showing the second step (after the initial Stock Concentration calculation) in a more involved example.

5.3.3. Technical Infrastructure

This particular implementation of the Drugulator is accessed from a website and therefore adopts predominantly a client-server model as shown in Figure 12. An *Apache* web server initially handles general requests from the Drugulator website before passing Drugulator-specific inputs on to a *Tomcat* server. This server, in turn, transfers the inputs to *Mathematica* (via a *webMathematica* application) which then calls a *Drugulator* package that evaluates the mathematical expression. Hence, in this model, the mathematical expression is first constructed within the client (using *Javascript* within an Internet Explorer Browser) before being sent to a computational back-end (a *Mathematica* kernel) for evaluation. The development environment used to develop the web *webMathematica*, *Javascript* and *Mathematica* code (of the *Drugulator* package) was the *Wolfram Workbench* with version control provided by the *Subversion* plug-in.

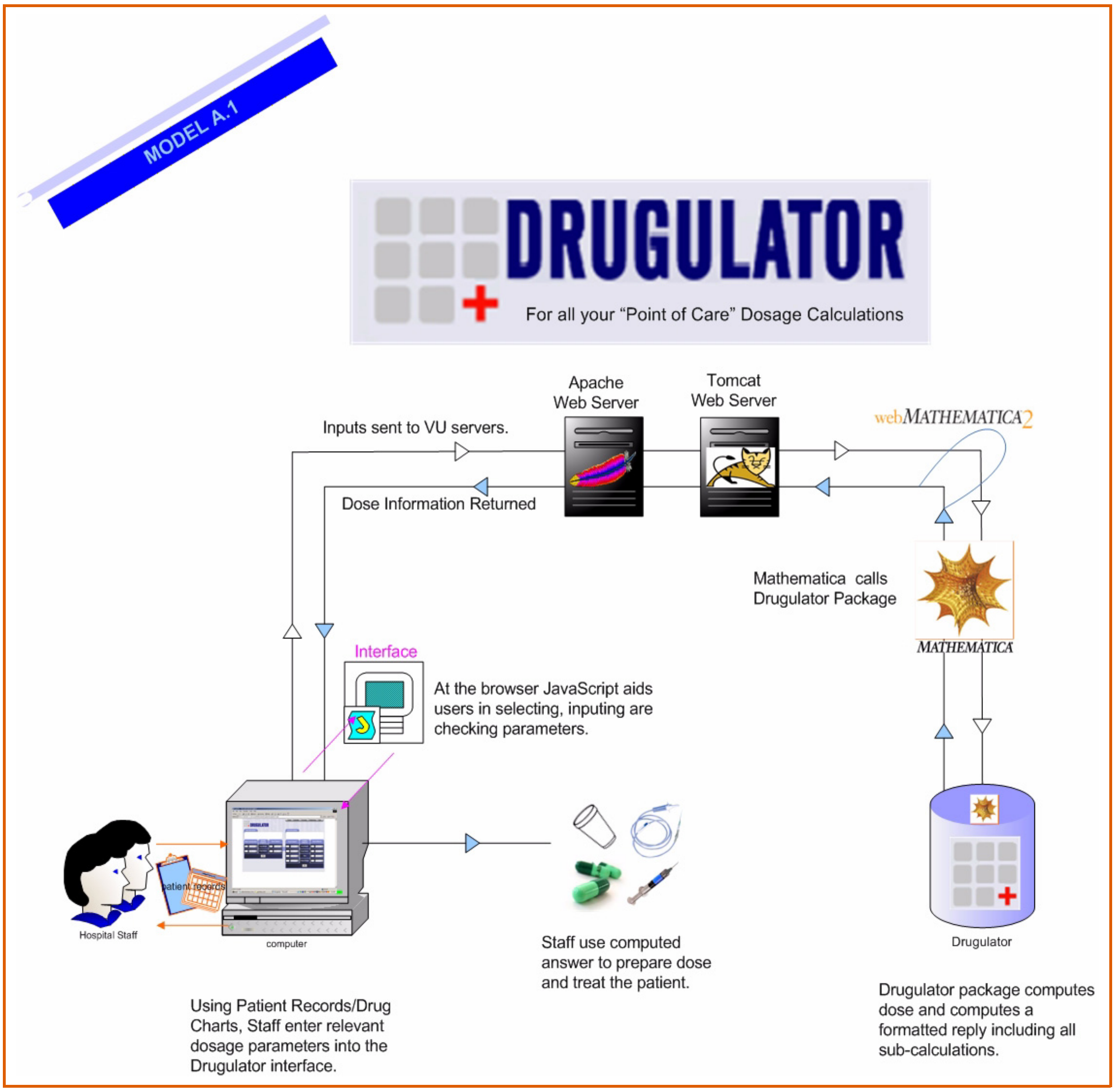


Figure 12: The architecture of first Drugulator implementation connecting a (Javascript-enabled) web-browser with a (Mathematica) computational backend.