clados Calculator

User's Guide

2005/07/12

Rough Draft

Interworld Transport

Dr Alfred W Differ

adiffer@gmail.com

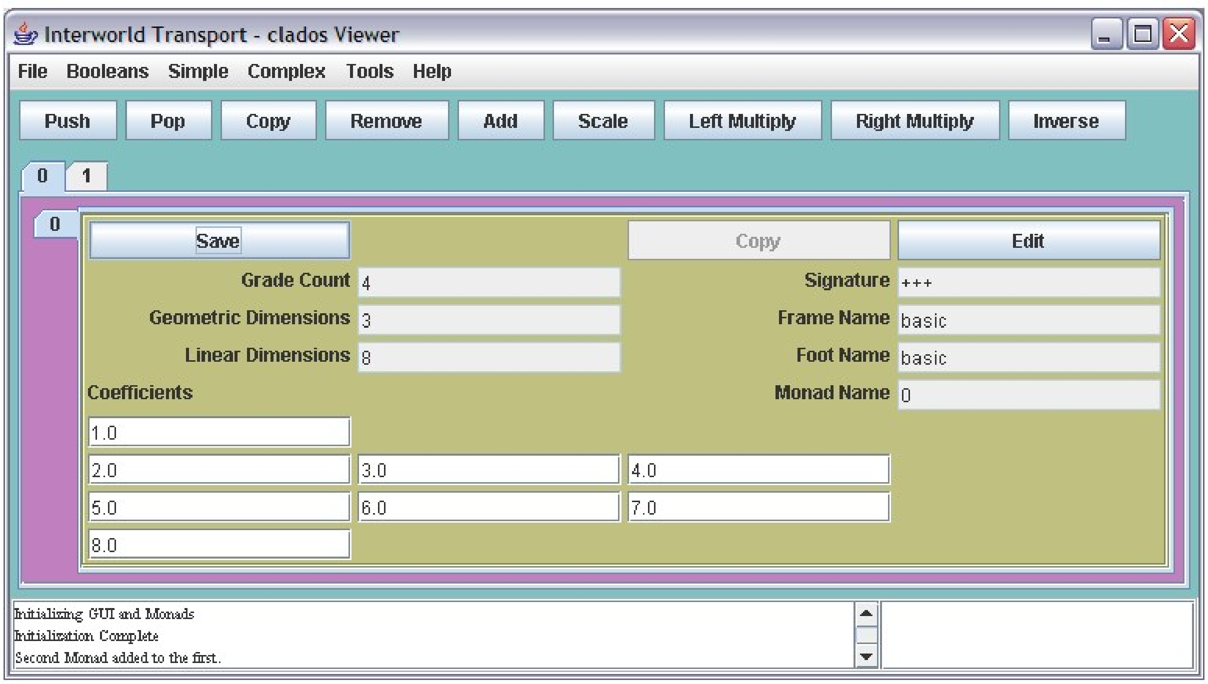


Table of Contents

Introduction

The clados calculator is a dual purpose application. On one hand it help developers understand what the clados library contains and how its parts work. On the other hand it helps students work with geometric algebras by providing them with the equivalent of a simple graphical calculator. For both purposes, the background machinery is hidden since it is not necessary to know exactly how some operations work and how some concepts are represented. Students will encounter such things elsewhere in their studies and software developers usually want to avoid them.

The calculator intentionally does <u>not</u> attempt to perform a number of functions that are already better represented in other similar applications.

- Graphical Representation
- Scripted Calculations
- Typesetting Functions

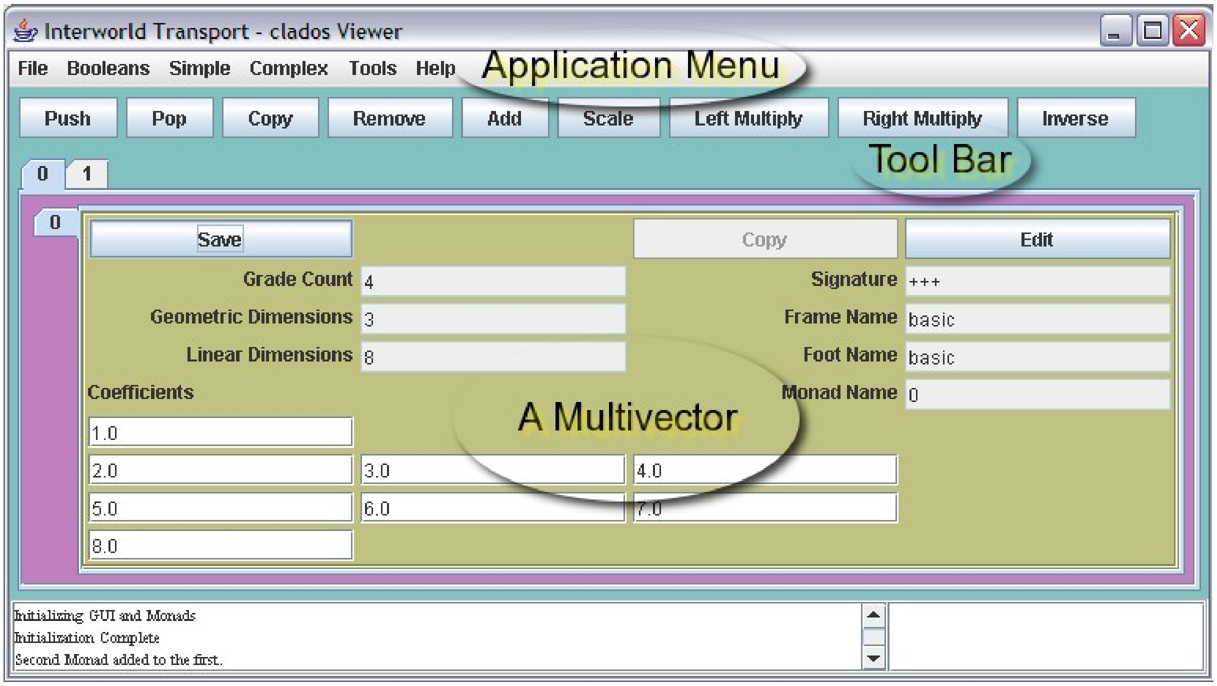
Graphical representations of the geometry described by multivectors are relatively straightforward for algebras with a small number of generators. However, the development work quickly escalates for higher algebras and mixed signatures. Development of such a feature would turn the application from a calculator to a presenter since the bulk of the code would be for drawing the geometry. Others may tackle such an effort if they wish for higher algebras, but we won't here. There are already good tools available for the simpler cases.

Scripted calculations involve the development of a suitable programming or macro language. Well developed presentation of the objects and results for use in other papers and publications involves suitable integration with adapters for each external tool. This author has little interest in such things, but is open to any work others would offer in these directions. One exception to this is the author's intent to develop a simple save and load routine so the current objects in the stack may be stored and reloaded at a later time.

The clados calculator on its face is a simple calculator much like electronic calculators with four buttons on them for addition, subtraction, multiplication, and division. We need a few more buttons to handle stack operations and the fact that multiplication isn't commutative most of the time, but the level of complexity is about the same. The calculator operates in the reverse polish sense (RPN) when performing operations requiring more than one multivector, so the stack operations might be familiar to some users. It also has the ability to answer a few basic questions using the boolean operations and save some steps with other simple operations. When the user is ready for more complicated, compound operations that go beyond the four-functions style, they can be found on the menus.

The User Interface

The face of the calculator is divided into a number of areas to group certain features together.



Status Text

In the lower left of the screen is a text area where the calculator shows text information about recent operations. If you add two multivectors successfully, text will be shown demonstrating that fact. If there are errors to present to you, they are show here too. Some operations find information related to the multivector and need a way to show it to you. They use this same text area to do it.

The status text area has a scroll bar you may use when the number of messages grows large. You may also remove the messages by selecting them and then deleting them whenever you wish.

Number Input and Results

In the lower right of the screen is a small text area that accepts and reports numbers. Some operations on a multivector require number inputs or produce number outputs. Inputs are drawn from this area and some outputs go there too. This region is much like the number display on a regular calculator except that it is also capable of accepting text in case the calculator ever needs it.

Tool Bar

The tool bar is the row of buttons near the top of the screen. The simpler operations are accessible from these buttons. The four buttons on the left help to manage to stack of multivectors while the remaining five act upon the top multivector in the stack and may make use of the second one if necessary.

Multivector Panes

Multivectors are displayed using tabbed panes near the center of the screen. The tabs that run along the top from left to right show which multivector is being displayed at the moment with the left most tab showing the one at the top of the stack. The next tab shows the multivector in the second slot in the stack and the right most tab shows the one at the bottom of the stack. If multiple rows of tabs are ever shown, use the push and pop buttons to figure out the relative order of two multivectors in the stack.

Buttons that apply to a multivector whether it is at the top of the stack or not are located on the pane with it. Some are disabled at present and represent future function that will be available when the calculator can support multivector lists (nyads).

Each tab holds one multivector at present. The fields on the tab show copies of the details normally hidden inside. Altering information in one of the fields does <u>not</u> alter the underlying multivector until the user clicks the 'Save' button on the same tab. Other tabs have their own 'Edit' and 'Save' buttons, so making a change on one tab and clicking a 'Save' button on another tab will <u>not</u> save the first changes. The user must use the correct 'Save' button to make the changes stick.

Application Menu

The application menu holds entries for all the operations that may be performed with the calculator including the ones present on the tool bar except for the stack managing operations. Most of the more complicated operations and all of the information seeking operations can only be performed from the menus.

Other helpful features of an information nature are also available from the application menu. Look for future features to be added here.

Managing and Creating Objects

Creating numbers for a regular calculator is usually just a matter of keying them in using the buttons. Multivectors are a little more complicated since several numbers are involved along with a few text fields to help define the kind of algebra being used. The clados calculator has two ways to create new objects. The first allows to you copy existing ones while the second one lets you start fresh.

Pushing and Popping

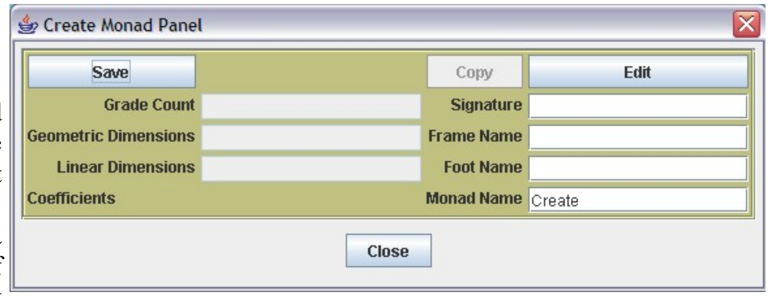
Clicking the 'Push' button will take the currently selected tab and swap it with the one to its immediate right if one exists. The 'Pop' button swaps with the tab to the left if it exists. In both cases, the names on the tabs are kept with their corresponding multivectors. A few pushes and pops, therefore, can result in a jumbled order of tab names.

Copying

Copying a multivector places a new multivector on a pane at the end of the stack. Pick the one you want to copy by selecting the tab for it. Click the copy button on the tool bar and a new version of the multivector will be added to the end of the stack. All things about the first multivector are copied including the name, signature, and coefficients.

If you are working with many multivectors from the same algebra, then the copy button will be the fastest way to make them. Once one exists in the stack, many copies can be made quickly. Each copy can then be altered a bit by clicking its 'Edit' button to open all fields and clicking 'Save' to keep all the changes.

Starting Fresh

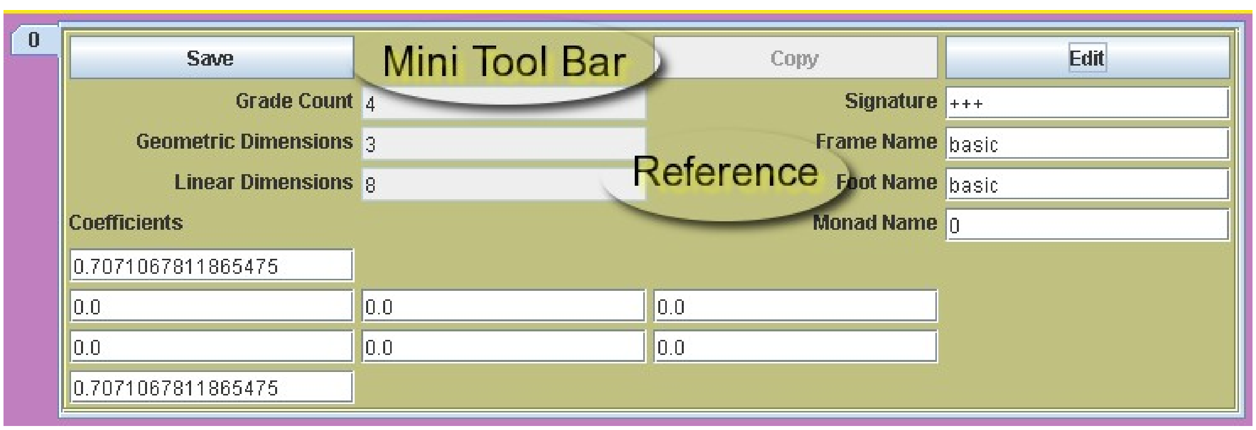


Creating new multivectors is accomplished by choosing 'Create Monad' menu from the Tools menu. A new screen appears to accept input for the top most fields in a multivector. Once a user fills them in and clicks save, a new multivector will be added to the end of the stack with zero coefficients. Go to that tab and edit the coefficients if a non-zero multivector is desired. Leave the name of the monad as 'Create' for this screen and edit it later if desired.

Creating new multivectors is a two step process. The number of coefficients is not known until the type of algebra is chosen. The calculator enforces this two step operation by using a limited input screen and then allowing the user to edit the coefficients later.

A Multivector

An element of a Clifford algebra is a multivector. The clados package actually implements a class called a 'monad' instead of a multivector. A monad is a multivector along with some other information to keep track of the reference frame against which the coefficients are measured and the algebra against which the product operation is defined. The name difference is largely cosmetic, though it is useful in avoiding name space collisions with other Java libraries maintained by other authors.



Mini Tool Bar

The mini tool bar holds two buttons that connect the fields a user sees on the screens to the multivector hidden behind. They give the application a convenient way to know when the user wants to affect alterations to that multivector. As such, they are not really part of the multivector, but it is handy to place them nearby.

Reference Information

The text fields on the right represent the reference information for a multivector. They must be defined before any coefficients may be added to the multivector.

- The signature field shows the squares of the generators of the algebra. It is a string of '+' and '-' signs strung together and is one of the first things that must be determined when establishing a new multivector in the calculator. At present, the calculator lets you think you can change the signature of a multivector, but the change won't stick. The fact that the field becomes editable is so that the copy function can do some things that would otherwise be blocked.

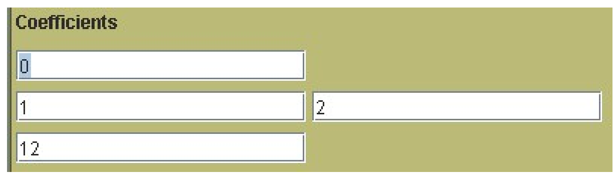
- The frame name is a simple string to name the reference frame used to define the coefficients. Frame names often reference their origins, but they might describe other properties of the frame too like whether or not it is inertial, whether or not it is right-handed, and so on.
- The foot name is a simple string to name the point in the algebra where the coordinates are 'tangent' to an underlying curved manifold. If one deals with strictly flat manifolds, there is no need to track the foot location. However, in curved spaces, it is useful to know where the tangent point is so one does not unintentionally perform illegal operations. Another use for the foot is to name the whole algebra.
- The monad name is a simple string the user may assign to the multivector with no consequences at all. It is there to help the user distinguish one object from another.

Coefficients

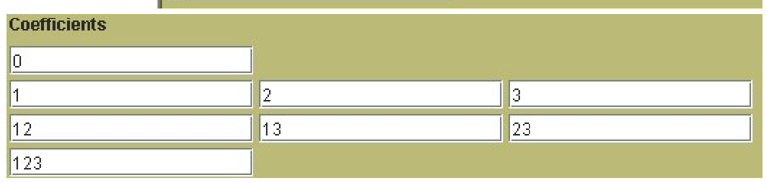
The coefficients are arranged to have the grade they represent increase as one works from top to bottom. The first and last ones represent scalars and pseudo-scalars respectively. The second row is for grade one, the third row is for grade two, and so on.

For grades where more than one coefficient is present, the user needs to know which coefficients go where. A few examples will suffice to show the pattern, though, so the calculator does not display them directly.

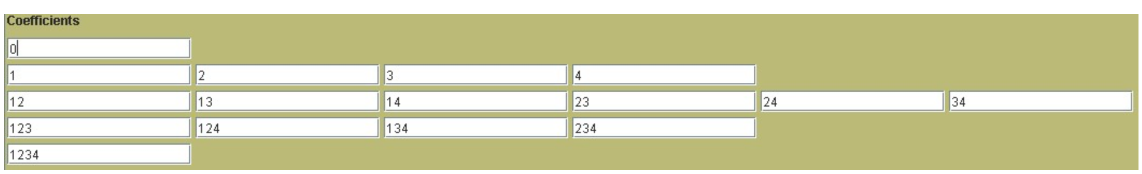
- In a two generator algebra, the grade-1 elements are arranged in the same way as the signature of the algebra. If the signature is '+-', the second field shows the element with the negative square.



- In a three generator algebra the grade-1 elements are arranged as usual. The grade-2 elements start with the first and second grade-1 elements and then work through all possible pairs in ascending order for the second element until there are no more. The next element picks up with the second and third grade-1 elements and then work through all possible pairs exactly like the first group.



- In a four generator algebra, the grade-1 and grade-2 elements are arranged as usual. The grade three elements start by pairing the first grade-2 element with the first grade-1 element not already contained in the grade-2 element. All possible triplets are arranged in ascending order of the last element and then things pick up again with the next available grade-2 element.



Once the user has a little practice with the multivectors they typically use, they won't need visual cues to know which field is which.

Extras

Above the coefficients are three extra fields that are never editable. They show the grade count, number of generators for the algebra, and the number of linearly independent elements in the corresponding vector space.

Naming the signature of the algebra is sufficient to set these fields, so the user never needs to alter them. They are shown as conveniences for now, but may be hidden at a later date.

Boolean Operations

The operations in this group all answer questions about the multivector at the top of the stack. Some use number inputs and others use the next object in the stack. All of them, though, answer questions that can be answered with a 'true' or 'false' statement. Look to the status text area for the answers.

Grade Checks

The 'is Grade?' operation takes an integer as its input and then checks the multivector to see if it has a pure grade of the same number.

- Zero multivectors always test true for any grade less then or equal to their grade count.
- A test for a grade that is too high for the multivector will return an error.
- A test on a multi-grade multivector will report a false result by saying it is not a pure grade match.

The 'is Multigrade?' operation does not require an integer input. It reports true if more than one grade is present in the multivector and false if not. Remember that zero multivectors report as being of pure grade.

Reference Check

This operation looks at the second multivector in the stack and checks to see if the signature, frame name, and foot names match the corresponding fields in the first multivector. If they don't, a false report is returned. If they do, a true result is returned.

This operation is unlikely to be used directly by users of the calculator, but other operations make use of it behind the scenes. For example, addition of two multivectors is blocked if they do not pass the reference match test. This is done to avoid having operations performed on objects that are not derived from the same algebra and rendered in the same reference frame.

Zero and Equal

The zero match operation looks at the first multivector in the stack and checks to see if all the coefficients are zero. If they aren't, a false report is returned. If they are, a true result is returned.

The equal operation tests the parts of the first and second multivectors in the stack to see if they are the same. For them to be the same, the coefficients must match and the reference test must pass.

Neither of these operations is likely to be used directly by a user. They are present because of the way Java handles nulls and equality on an object instance level.

No Inverse

Some algebras contain elements that do not have a multiplicative inverse. Nilpotents and idempotents are special cases one must be able to detect. Just as one must be careful to avoid dividing a number by zero, handling nilpotents (objects with a vanishing square) and idempotents (objects that are their own squares) requires special care. Multiples of idempotents are also special cases; so three tests are included in the calculator and the monad class to help spot them.

- The 'is Nilpotent?' operation reports true if the object has a zero square even if the original object would also pass the zero test.

- The 'is Idempotent?' operation reports true if the object doesn't change after being multiplied by itself.
- The 'is Multiple Idempotent?' operation returns true if the object only changes by a scale after being multiplied by itself.

Other Operations

Many other operations are available within the calculator. The first group under the 'Simple' menu uses only the top multivector in the stack. The second group under the 'Complex' menu uses the first two multivectors in the stack. All operations are performed in a manner that alters the multivector on the top of the stack.

Magnitudes

Two operations report on the magnitude of a multivector. The first delivers the magnitude while the second delivers the square of the magnitude. Both operations are present because some developers may wish to avoid the computational overhead of the square root operation when they face situations where the square of the magnitude is sufficient. Results are reported to the input/output field in the lower right.

Grade Alterations

Two operations alter the multivector at the top of the stack by keeping or suppressing a particular grade. The both require an integer input from the input/output field in the lower right of the screen. The 'preserve Grade?' operation will suppress all other grades and the 'suppress Grade?' operation will suppress only the one grade described by the offered integer.

Inverse and Normalization

Some multivectors can be normalized and inverted in the multiplication sense. These two operations perform the tasks on multivectors when it can be done at all. Neither operation returns any results except status information. They just alter the multivector at the top of the stack if they can.

Reverse and Invert

The active 'Invert' operation changes all the generators in the algebra to their negatives and then calculates the impact that has on the multivector. Eventually a passive version of this operation will be added.

The 'Reverse' operation changes the order of multiplication on all elements. A person who read from right to left would build their higher rank objects in a reversed manner compared to one who reads left to right, so this operation switches between the two methods. After the swap is done, though, the impact of the alteration is calculated and represented in a left-to-right system again.

Dual

The local dual operation multiplies the multivector at the top of the stack by a normalized pseudo-scalar on the left and then calculates the result. The effect tends to swap low grade coefficients for high grade coefficients. Duals are quite handy to recover results associated with cross products in three dimensions and author-based definition choices in higher algebras.

Scale

The scaling operation simply multiplies the multivector by the number provided in the input/output field in the lower right of the screen. This alters all coefficients equally since each one of them is multiplied by the same number. The status text will report what the scale factor is even though it is present in the input/output field.

Add and Subtract

The addition operation will add the second multivector in the stack to the first one. The subtract operation will subtract the second one from the first. Both operations will be blocked if the reference match test involving the two of them doesn't pass.

Multiplication

Left-side and right-side multiplication are managed with separate operations. In the first case, the second multivector in the stack is multiplied against the first one from the left. The right-sided operation does the same thing except the second multivector is multiplied from the right. Both operations will be blocked if the reference match test involving the two of them doesn't pass.

Dots and Wedges

In some cases, the two top multivectors can participate in dot and wedge products. If they are defined at all, they are symmetric and antisymmetric products of the two, so both of these operations provide the user with a quick way to perform a more complicated set of steps.

More to Come

Other disabled operations are visible on the menus. They will be enabled as they are completed in each of the underlying classes in the clados library. Each will have restrictions on the second multivector and a set of error messages to help make their usage clear.

Glimpse of the Future

Some changes are already planned for the calculator. Others can be added if enough users care to want them or help write and test them. Most of the changes that are currently planned are associated with changes to the monad class and the new nyad class.

Save and Reload

The calculator currently starts with two zero monads from a +++ algebra over reals. Users can add new monads as they wish, but they can't yet save a new starting position. In the near future the application will be altered to save and read a text file containing the monads for display in the calculator.

Another minor change will be a configuration line allowing a user to start the calculator with monads from an algebra with a user defined signature.

Cosmetics

Some presentation choices for the user interface work better for monads in small algebras, but not ones in larger algebras. Configuration lines will be added to give the user some control over how it presents the objects it displays.

Multivector Lists (Nyads)

The multivector tabbed panes hold an extra tab with a zero label on all of them. This may strike some as strange, but it makes more sense once one realizes that a future version of the calculator will permit other monads to be added on higher numbered vertical tabs. This will make each of the horizontal tabs in the stack hold monad lists instead of single monads. These monad lists will appear in the clados library under a class named 'nyad.'

Motivation for this expansion comes from the demands of physics theories. Descriptions of simpler theories may be done in single algebras, but more complex theories with non-scalar properties tend to rely upon two algebras at a minimum. Special rules relate the algebras used so nyads must do more than present monad lists. New operations will probably be added and old ones redefined.

Development Help

Work on this calculator and the library it demonstrates is performed as part of an open source project. The code and documentation for it all are freely available to anyone who cares to have a copy.

Development and testing of the source code and documentation are just as open. If anyone wishes to jump in and participate on any level, his or her help will be welcomed. Expertise is not required as long as one has a willingness to learn.

Here are some of the skills we know we could use. If you have a skill that you think could be used that isn't listed, speak up anyway. It is possible we simply didn't think about adding it.

- Java code writing and testing
- Technical writing and editing
- Users willing to test things after learning how things 'should' work.

Appendix A – License Terms

DESIGN SCIENCE LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. PREAMBLE.

Copyright law gives certain exclusive rights to the author of a work, including the rights to copy, modify and distribute the work (the "reproductive," "adaptative," and "distribution" rights).

The idea of "copyleft" is to willfully revoke the exclusivity of those rights under certain terms and conditions, so that anyone can copy and distribute the work or properly attributed derivative works, while all copies remain under the same terms and conditions as the original.

The intent of this license is to be a general "copyleft" that can be applied to any kind of work that has protection under copyright. This license states those certain conditions under which a work published under its terms may be copied, distributed, and modified.

Whereas "design science" is a strategy for the development of artifacts as a way to reform the environment (not people) and subsequently improve the universal standard of living, this Design Science License was written and deployed as a strategy for promoting the progress of science and art through reform of the environment.

1. DEFINITIONS.

"License" shall mean this Design Science License. The License applies to any work which contains a notice placed by the work's copyright holder stating that it is published under the terms of this Design Science License.

"Work" shall mean such an aforementioned work. The License also applies to the output of the Work, only if said output constitutes a "derivative work" of the licensed Work as defined by copyright law.

"Object Form" shall mean an executable or performable form of the Work, being an embodiment of the Work in some tangible medium.

"Source Data" shall mean the origin of the Object Form, being the entire, machine-readable, preferred form of the Work for copying and for human modification (usually the language, encoding or format in which composed or recorded by the Author); plus any accompanying files, scripts or other data necessary for installation, configuration or compilation of the Work.

(Examples of "Source Data" include, but are not limited to, the following: if the Work is an image file composed and edited in PNG format, then the original PNG source file is the Source Data; if the Work is an MPEG 1.0 layer 3 digital audio recording made from a WAV format audio file recording of an analog source, then the original WAV file is the Source Data; if the Work was composed as an unformatted plaintext file, then that file is the Source Data; if the Work was composed in LaTeX, the LaTeX file(s) and any image files and/or custom macros necessary for compilation constitute the Source Data.)

"Author" shall mean the copyright holder(s) of the Work.

The individual licensees are referred to as "you."

2. RIGHTS AND COPYRIGHT.

The Work is copyrighted by the Author. All rights to the Work are reserved by the Author, except as specifically described below. This License describes the terms and conditions under which the Author permits you to copy, distribute and modify copies of the Work.

In addition, you may refer to the Work, talk about it, and (as dictated by "fair use") quote from it, just as you would any copyrighted material under copyright law.

Your right to operate, perform, read or otherwise interpret and/or execute the Work is unrestricted; however, you do so at your own risk, because the Work comes WITHOUT ANY WARRANTY -- see Section 7 ("NO WARRANTY") below.

3. COPYING AND DISTRIBUTION.

Permission is granted to distribute, publish or otherwise present verbatim copies of the entire Source Data of the Work, in any medium, provided that full copyright notice and disclaimer of warranty, where applicable, is conspicuously published on all copies, and a copy of this License is distributed along with the Work.

Permission is granted to distribute, publish or otherwise present copies of the Object Form of the Work, in any medium, under the terms for distribution of Source Data above and also provided that one of the following additional conditions are met:

(a) The Source Data is included in the same distribution, distributed under the terms of this License; or

(b) A written offer is included with the distribution, valid for at least three years or for as long as the distribution is in print (whichever is longer), with a publicly-accessible address (such as a URL on the Internet) where, for a charge not greater than transportation and media costs, anyone may receive a copy of the Source Data of the Work distributed according to the section above; or

(c) A third party's written offer for obtaining the Source Data at no cost, as described in paragraph (b) above, is included with the distribution. This option is valid only if you are a non-commercial party, and only if you received the Object Form of the Work along with such an offer.

You may copy and distribute the Work either gratis or for a fee, and if desired, you may offer warranty protection for the Work.

The aggregation of the Work with other works that are not based on the Work -- such as but not limited to inclusion in a publication, broadcast, compilation, or other media -- does not bring the other works in the scope of the License; nor does such aggregation void the terms of the License for the Work.

4. MODIFICATION.

Permission is granted to modify or sample from a copy of the Work, producing a derivative work, and to distribute the derivative work under the terms described in the section for distribution above, provided that the following terms are met:

(a) The new, derivative work is published under the terms of this License.

(b) The derivative work is given a new name, so that its name or title cannot be confused with the Work, or with a version of the Work, in any way.

(c) Appropriate authorship credit is given: for the differences between the Work and the new derivative work, authorship is attributed to you, while the material sampled or used from the Work remains attributed to the original Author; appropriate notice must be included with the new work indicating the nature and the dates of any modifications of the Work made by you.

5. NO RESTRICTIONS.

You may not impose any further restrictions on the Work or any of its derivative works beyond those restrictions described in this License.

6. ACCEPTANCE.

Copying, distributing or modifying the Work (including but not limited to sampling from the Work in a new work) indicates acceptance of these terms. If you do not follow the terms of this License, any rights granted to you by the License are null and void. The copying, distribution or modification of the Work outside of the terms described in this License is expressly prohibited by law.

If for any reason, conditions are imposed on you that forbid you to fulfill the conditions of this License, you may not copy, distribute or modify the Work at all.

If any part of this License is found to be in conflict with the law, that part shall be interpreted in its broadest meaning consistent with the law, and no other parts of the License shall be affected.

7. NO WARRANTY.

THE WORK IS PROVIDED "AS IS," AND COMES WITH ABSOLUTELY NO WARRANTY, EXPRESS OR IMPLIED, TO THE EXTENT PERMITTED BY APPLICABLE LAW, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

8. DISCLAIMER OF LIABILITY.

IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS WORK, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

END OF TERMS AND CONDITIONS