Conformiq Creator and Conformiq Transformer
Conformiq is undoubtedly one of the leading vendors of automated test case solutions.

Authors
Philip Howard
and Daniel Howard
Testing is expensive. Frequently, very expensive. It is often the case that testing a new or revised application is more expensive than the actual development of the application. On the other hand, if you don’t test properly you can end up losing business, delaying the introduction of new products, upsetting your customers and, potentially — if you are in a heavily regulated industry — you run the risk of fines if your systems fail to meet compliance and regulatory standards. So, you may think of it as a necessary evil, but the key word is necessary.

Given that testing is necessary but costly, it is important that testing environments be as productive and efficient as possible. In the past — and still today in all too many cases — testing has tended to rely on manual methods that require the employment of many people. Not only is this expensive, it is also error-prone, which itself has associated costs. As a result, modern trends are to automate as much of the testing process as possible and certainly those tasks that are relatively routine. Conformiq’s tools are designed to do just that.

Fast facts
Briefly, the way that Conformiq works is that you initially create a graphical representation, (that is, a model) of the system under test. This model is represented in an activity diagram which may be created in one of two ways: either by directly dragging and dropping objects onto a palette (including bringing in details from a complementary structure diagram) or by automatically creating the activity diagram by reverse engineering existing test cases, BPMN drawings, Gherkin user stories and so on. Once the initial activity diagram is created it can be synchronised with requirements brought in from third party environments with which Conformiq integrates or you can create a high-level requirements model directly within Conformiq Creator. Once complete, you select the coverage option most suitable for your project and automatically generate optimised executable test scripts, test validations, and test data, which can be automatically executed via third party test frameworks.

This automation is at the heart of Conformiq’s proposition: that you should never have to fall back on manual coding. As far as we know, the company’s product suite is unique in providing this capability: it generates executable test scripts, test validations, and test data directly from the model of the system under test. Conformiq Creator generates tests scripts that are ready for automated execution and Conformiq Transformer automates the link to the execution tool.

Key findings
In the opinion of Bloor Research, the following represent the key features of the products offered by Conformiq:

- We are especially impressed with how well integrated Conformiq is with third party testing and development environments and associated products. The products should slip easily into existing environments, regardless of whether these are in-house or outsourced.

- Sub-models are supported so that large and complex applications can be broken down into more manageable parts. This will also be useful in supporting agile development practices.

- A major focus for the company is in ensuring that testers have both visibility into, and control over, the processes involved when using Conformiq.

- We very much like the way that Conformiq Transformer can generate an activity diagram from a suite of existing test cases. Reverse engineering of existing test assets automatically identifies redundant tests and subtests. The software is also able to resolve logical inconsistencies within existing tests. Where this is not practical users can graphically review and find deviations from expected application behavior and then do integration testing of the new design enhancements with the existing application.
If you import requirements from a third-party tool, then Conformiq provides requirements traceability across its platform. Alternatively, if you do not import requirements then Conformiq provides traceability back to its system model. However, in the latest release there are new modelling capabilities, targeted at business analysts, that will allow them to create a simple requirements model, in which case requirements traceability will be extended further.

Unlike some other companies that simply generate test cases that you can subsequently optimise, Conformiq generates test cases that are optimised from the get-go.

The complementary Conformiq Grid moves test generation to the cloud and should provide significant performance advantages for complex models.

**The bottom line**

Conformiq is all about automating everything from requirements to test execution. Automation is more than just execution, it is design, generation and execution. The truth is that there are other tools that automate test case generation but none that does it as fully automatically for directly automated execution. Conformiq argues that it removes manual activity from the testing process, which it does. However, it does this at the expense of more complex modelling. With other tools you might have to write code snippets to enable test execution and when the system under test changes you might very well have to amend that code. But in Conformiq you amend the model: which approach involves more effort is a question we can’t answer.

Ultimately, selecting Conformiq for test automation is a philosophical decision: if you like the company’s approach of driving everything, in a completely automated fashion, from a comprehensive model, then Conformiq really shines.

“Automation is more than just execution, it is design, generation and execution but in many ways it is the company’s support for testers that is most significant.”
The products

Conformiq offers four products: Conformiq Creator, Conformiq Transformer, Conformiq Designer and Conformiq Grid. We are focusing on the first two of these in this paper. Conformiq Designer is similar to Conformiq Creator but is designed for testing embedded systems rather than free-standing enterprise or IT applications. It is organised around the use of UML-based state diagrams rather than the activity diagrams used by Conformiq Creator. Conformiq Grid is a cloud-based offering parallelised test generation by running across multiple CPUs to support either Conformiq Creator or Conformiq Designer.

Conformiq Creator and Conformiq Transformer are in version 3.1 and 1.4 respectively. The latter was only released at the beginning of 2016. Both products run on either Windows or Linux platforms and they are Eclipse based.

The Conformiq suite of tools can be used to test applications of many different types including UI, desktop, API, web services, back office, mainframe, mobile, packaged applications, and combinations of these.

Requirements integration

The first thing you need to understand about Conformiq Creator and Transformer is that they’re not intended to be used alone. Although of course that is possible, they’re clearly meant to be used together, where Creator is used to automate the generation of test cases before Transformer is used to automate their execution. And although you could use the two of them (or even a single one of them) as a stand-alone product, they’re meant to be used as part of a mature testing solution. In fact, the extensive capacity for integration with other tools is one of their biggest selling points.

It is therefore no surprise that requirements integration is a major component of how Creator is intended to work. Requirements are imported directly from a variety of requirements management tools, BPMN (business process model and notation) supporting products, or Excel and are then ready to be added to the model (see the next section). This is actually very simple: you are presented with a list of your requirements and you drag-and-drop them onto your model.

Creator also has the capacity to create requirements from whole cloth. However, these ad hoc requirements cannot be exported into any sort of external requirements tool. By design, communication is one way so that each tool’s responsibilities are properly encapsulated. For the same reason, you cannot edit requirements imported from an external tool. However, this does mean that unlike imported requirements, any requirements created in Creator lack any sort of traceability and for this reason creating requirements directly in Creator is not recommended.

Model-based testing

Although requirements are an important part of Creator, the heart of a Creator project is the model, a graphical representation of the expected operation of your application, API, or a combination of technologies that are to be tested using your testing environment, as shown in Figures 1 and 2.

This takes the form of an activity diagram and can be generated in several different ways. First, and most obviously, it can be created manually inside Creator. Second, it can be imported from another tool (such as Microsoft Visio, Gherkin feature files, Micro Focus BPT, or Excel) that can create activity diagrams. Last, and certainly not least, the model can be
generated from a collection of existing test cases. This last feature is very impressive and bears exploring in more detail.

The application is obvious: any company worth its salt will already have a collection of test cases (usually a huge number of them). Automatically generating a model from these test cases represents a potentially enormous saving in terms of time and cost. Unfortunately, this approach is not without its downsides. Many companies have, as pointed out before, a huge number of test cases and being able to select exactly those which are up-to-date and correct is most definitely not a given. Even though Conformiq operates under a ‘garbage in, garbage out’ paradigm, errors related to redundant tests and subtests are identified as test suite consistency errors and Creator automatically optimises out those tests.

If there are incomplete or incorrect test cases, Conformiq relies on – and recommends as best practice – domain experts examining the resulting models and commenting on their correctness.

As for the activity diagram itself, this will be familiar to anyone who’s previously worked with the Unified Modelling Language (UML). The way that this works is that you create an initial model that lacks details about test data and test objects. These are separately modelled using a structure diagram and then imported into the activity diagram where the model is extended in an appropriate fashion to capture the actions and flow of the application. When you load requirements these are also dragged and dropped onto your palette to further enrich the activity diagram, and to ensure that all requirements are covered and are identified as being covered in the generated test cases. The result is very comprehensive – indeed, this is a general strength of Conformiq – and implementing it in stages makes the creation of your model simpler than would otherwise be the case. This is especially true in the latest release as the company has introduced a new modelling facility that enables Business Analysts and other team members to create a simple business requirements level model of the system’s operation. This can be updated as it evolves during the agile development process. It is also possible to have models call other models (in effect, supporting sub-models) thus further offering the possibility of simplification. Figure 1 shows an example business requirements level model.

Nevertheless, any particular activity diagram that has been created to support automated execution (rather than capturing business process logic) remains relatively complex, because that level of detail is required to automatically design and generate the test scripts, validations, and data for automated test execution. Figure 2, for example, shows an extremely simple system – a login with no real error handling – and is still rather technical, involving not just the actual program flow (the login itself) but also user input data (with corresponding data types) and expected data. Any experienced developer or tester should be able to interpret such a diagram easily enough (though they may need training before creating it) but it would probably need to be explained to any business user with whom you wanted to collaborate. This is the reason that Conformiq has introduced a higher level, business-oriented model in the latest release, as previously discussed.

**Test generation and automation**

Once you’ve got your requirements, your model, and you’ve added the former to the latter, it’s time to actually generate your test cases. This is straightforward: the only thing you need to do first is to decide which type(s) of coverage you’re going to use.
Creator offers an extensive choice of coverage, each of which can be used either individually or together with other types of coverage. They are:

- **Requirements Coverage**: cover "requirement" statements.
- **Condition Coverage**: cover "true" and "false" cases of conditional constructs.
- **Risk Based Coverage**: cover user assigned priorities.
- **Data Coverage**: cover all pairs or all combinations of data values.
- **Boundary Value Analysis**: cover integer boundary conditions.
- **All Paths – Control flows**: cover all arbitrary action sequences.

Having selected one or more of these, Creator will go away and automatically generate the test cases. Once it’s done, you’ll be presented with a wealth of information about each test case, including traceability to each fulfilled requirement and a graphical representation of the test case’s route through the model, as shown in Figure 3 below, for known coverage and traceability if test cases fail during execution.

At this stage, you have several options for exporting these test cases, but for now we will focus on exporting into Creator’s sister product, Conformiq Transformer (the other options are covered in a later section).

Transformer is designed to automate test execution in the same way that Creator is designed to automate test design and, similar to Creator, it can be used as a stand-alone tool. In this case, the first thing you must do is configure the test automation. Transformer uses a keyword-based test automation system: you have a repository of 'keywords', where each keyword consists of a series of automation steps, and each step has a corresponding piece of programming code. Then, having created your test in Transformer, you can add a keyword or keywords to each step in your test case. When the test script is generated, it converts each step in the test case into a series of keywords, each keyword into a series of automation steps, and each automation step into programming code. This code is then assembled into the final test script for automated execution in Micro Focus UFT and/or Selenium. Further, manual test cases can be imported directly into Transformer and it will enable direct automated execution in either tool.

There are some additional features which are worth pointing out here. There is the capacity for reusing keywords: each keyword can have a number of variable parameters. This means that, for instance, you could use the keyword ‘Click button’; with a variable for the button name, in any test case to simulate clicking any button, regardless of what that button is called. In addition, although normally you would need a developer to write code for each automation step, in many cases this can be done for you via the built-in UI inspector. Basically, you activate the inspector, and it opens a web browser for you inside Transformer. Anything you do inside this browser is recorded in the inspector, and can subsequently be copy-and-pasted out of it to define an automation step with no coding required.

However, this is all assuming you use Transformer without Creator. With Creator, everything becomes much easier: you simply import your model into Transformer and all of your tests, keywords and automation steps are generated automatically. However, this process shares the same caveat as the automatic model generation in Creator: if your model isn’t defined properly, the items generated by Transformer won’t be either. But again, the ability to graphically visualise the expected operation of the system under test and do a peer review...
at an early stage should enable teams to discover errors, ambiguities, and omissions.

**Test case management**

In addition to the test automation and test script generation functions described above, Transformer also handles manual tests. You can import either manual tests or Gherkin feature files. These will be picked up by Transformer and transformed into automatically executable tests. In addition, any test can be exported as a manual test – this feature is available in multiple languages (English, French and German are supported by default).

Furthermore, Transformer automates your test execution by extending existing test automation for Selenium and UFT (in fact, Selenium is built-in). Test results are then stored and can easily be accessed inside Transformer. Together with Creator, all requirements from all requirement providers. The requirements list inside Creator will then be updated, showing you explicitly which requirements (if any) have been added or removed. This is shown in **Figure 4**.

The model must then be updated with the new requirements: usually, this will amount to adding any such requirements to the model. If the model itself needs changing, Creator will display error messages if any breaking changes are made.

Lastly, new tests may need to be generated – thankfully, doing this is easy. As before, you only need to tell Creator to start test generation and it will automatically analyse existing tests to see if they’re still valid, then proceed to generate new tests to cover any goals which were previously being covered by invalid tests.

**Integrating with test data and virtual services**

Test data is integral to automated testing and you define your test objects as part of a structure diagram as previously discussed. From this, the data required to design the test cases will be automatically generated from the logical operation of the model (this is why Creator models are of the system operation, not the test flows). User selected data will either live inside the model or in an external spreadsheet, and you can also import data via Excel. At this time, Conformiq does not directly integrate with any third party test data management tools. On the other hand, Conformiq does integrate with service virtualisation products – Parasoft, for example, is a partner, so that you can test against simulated and virtual data.

**Integrating with the broader DevOps environments**

As mentioned above, integration with other tools is one of Creator’s and Transformer’s biggest selling points. This doesn’t end with requirements, either: the array of tools which Creator and Transformer can integrate with is, frankly, very impressive.

You can import requirements from a variety of requirements management tools, as mentioned, and Conformiq also
integrates with products such as Micro Focus (formerly HPE) ALM, CA Rally, Atlassian JIRA, and a number of others. You can import existing assets in several different ways: Visio files and collections of manual tests (including those stored in Excel), XSD, and JSON files can be used to generate models; Gherkin feature files and manual tests can be imported directly or converted into automated tests.

Conversely, you can also export test cases and test scripts to a number of different tools. You can export test cases purely as documentation or manual tests in a spreadsheet or as a pdf or html file, in several different languages. You can export them from Creator to a dedicated test management tool, such as, again, Micro Focus ALM. Of course, you can also export them to Transformer, and from there, you can export the generated test scripts to an automatic test execution tool, such as Selenium, or a Jenkins server. You can also export scripts – thanks to Conformiq’s open API - to various automated execution tools including the aforementioned Micro Focus UFT, LeanFT or Mobile Center, Selenium, Appium, Tricentis Tosca, Robot, Parasoft, SoapUI, EggPlant, amongst others. The point here is that Creator and Transformer come with a wealth of integration options to fit them into almost any existing testing environment. Let’s be realistic: the vast majority of companies with any appetite for testing already use at least some of the tools mentioned above. What’s more, the above list is nowhere near complete. It’s safe to say that, whichever testing products you use, Conformiq can accommodate you.

…”integration with other tools is one of Creator’s and Transformer’s biggest selling points. This doesn’t end with requirements, either: the array of tools which Creator and Transformer can integrate with is, frankly, very impressive."
Conformiq is a privately owned company backed by venture capital. It was founded in Finland in 1998, initially as a testing services company focused on the telecommunications sector. Early clients were Ericsson and Nokia. The company now has its headquarters in California though research and development continues in Helsinki (and also in Bangalore). The company also has offices in Paris and, in conjunction with a support partner in India, is able to offer 24x7 support across the globe.

The company has a number of go-to-market partners, several of which have global reach in their own right. Technology partners include Micro Focus (formerly HPE), IBM, Parasoft, Tricentis, TestPlant, ARIS (Software AG), and Sparx Systems, and the company has also integrated its products with those of many other vendors.

It is worth noting that arguably the biggest competitor to automated test environments such as the one offered by Conformiq is simply to recruit bodies for manual testing. This has the perceived advantage that people represent operating expense rather than capital expense. To counter this, Conformiq – while it normally licenses its software in a conventional manner – is flexible enough to offer subscription and usage based charges, if required. This is a significant competitive advantage.

Website: www.conformiq.com
As a general principle we are very much in favour of introducing as much automation as possible into testing cycles. We are also fans of model-driven approaches to this. However, there are different ways in which this can be accomplished. There are vendors, for example, that do not expose their model at all (effectively an approach based on object modelling). Then again, there are suppliers whose solution is based on requirements modelling. And, of course, there is Conformiq, which operates from a system model. Thus, there is a spectrum of model-based offerings that are increasingly automated. As a rule of thumb, the more comprehensive the model the more automation, but this does come at the expense of model complexity. This is a cost that may be worth paying. If so, you should certainly be considering Conformiq.

FURTHER INFORMATION
Further information about this subject is available from www.BloorResearch.com/update/00002357
Philip started in the computer industry way back in 1973 and has variously worked as a systems analyst, programmer and salesperson, as well as in marketing and product management, for a variety of companies including GEC Marconi, GPT, Philips Data Systems, Raytheon and NCR.

After a quarter of a century of not being his own boss Philip set up his own company in 1992 and his first client was Bloor Research (then ButlerBloor), with Philip working for the company as an associate analyst. His relationship with Bloor Research has continued since that time and he is now Research Director, focused on Information Management.

Information management includes anything that refers to the management, movement, governance and storage of data, as well as access to and analysis of that data. It involves diverse technologies that include (but are not limited to) databases and data warehousing, data integration, data quality, master data management, data governance, data migration, metadata management, and data preparation and analytics.

In addition to the numerous reports Philip has written on behalf of Bloor Research, Philip also contributes regularly to IT-Director.com and IT-Analysis.com and was previously editor of both Application Development News and Operating System News on behalf of Cambridge Market Intelligence (CMI). He has also contributed to various magazines and written a number of reports published by companies such as CMI and The Financial Times. Philip speaks regularly at conferences and other events throughout Europe and North America.

Away from work, Philip’s primary leisure activities are canal boats, skiing, playing Bridge (at which he is a Life Master), and dining out.

Daniel started in the IT industry relatively recently, in only 2014. Following the completion of his Masters in Mathematics at the University of Bath, he started working as a developer and tester at IPL (now part of Civica Group). His work there included all manner of software and web development and testing, usually in an Agile environment and usually to a high standard, including a stint working at an ‘innovation lab’ at Nationwide.

In the summer of 2016, Daniel’s father, Philip Howard, approached him with a piece of work that he thought would be enriched by the development and testing experience that Daniel could bring to the table. Shortly afterward, Daniel left IPL to work for Bloor Research as a researcher and the rest (so far, at least) is history.

Daniel primarily (although by no means exclusively) works alongside his father, providing technical expertise, insight and the ‘on-the-ground’ perspective of a (former) developer, in the form of both verbal explanation and written articles. His area of research is principally DevOps, where his previous experience can be put to the most use, but he is increasingly branching into related areas.

Outside of work, Daniel enjoys Latin and ballroom dancing, skiing, cooking and playing the guitar.
Bloor overview
Technology is enabling rapid business evolution. The opportunities are immense but if you do not adapt then you will not survive. So in the age of Mutable business evolution is essential to your success.

We'll show you the future and help you deliver it.
Bloor brings fresh technological thinking to help you navigate complex business situations, converting challenges into new opportunities for real growth, profitability and impact.

We provide actionable strategic insight through our innovative independent technology research, advisory and consulting services. We assist companies throughout their transformation journeys to stay relevant, bringing fresh thinking to complex business situations and turning challenges into new opportunities for real growth and profitability.

For over 25 years, Bloor has assisted companies to intelligently evolve: by embracing technology to adjust their strategies and achieve the best possible outcomes. At Bloor, we will help you challenge assumptions to consistently improve and succeed.

Copyright and disclaimer
This document is copyright © 2018 Bloor. No part of this publication may be reproduced by any method whatsoever without the prior consent of Bloor Research.

Due to the nature of this material, numerous hardware and software products have been mentioned by name. In the majority, if not all, of the cases, these product names are claimed as trademarks by the companies that manufacture the products. It is not Bloor Research’s intent to claim these names or trademarks as our own. Likewise, company logos, graphics or screen shots have been reproduced with the consent of the owner and are subject to that owner’s copyright.

Whilst every care has been taken in the preparation of this document to ensure that the information is correct, the publishers cannot accept responsibility for any errors or omissions.