

professional
audio

professional **audio**

Das Magazin für Aufnahmetechnik



Praxis: DAW-Automation

Workshop DAW-Automation

So gelingt jeder Mix

Test: Microtech Gefell M 221

Mikrofon für Perfektionisten

Test: Zoom H2n

Surroundsound-fähiger Recording-Zwerg

Test: Elysia Xpressor

Lowcost-Kompressor mit Highend-Features

Interview

Mouse on Mars goes Classic

Test: Analog-Mischpult SPL Neos

Die ultimative Mix-Erfahrung

Test: Fostex HP-A3
Edel-Sound zum
Budget-Preis



brainworx 

...last nite an E

Zwischen 0 und 1

Hinter den Kulissen der Plug-in-Produktion, Teil 2

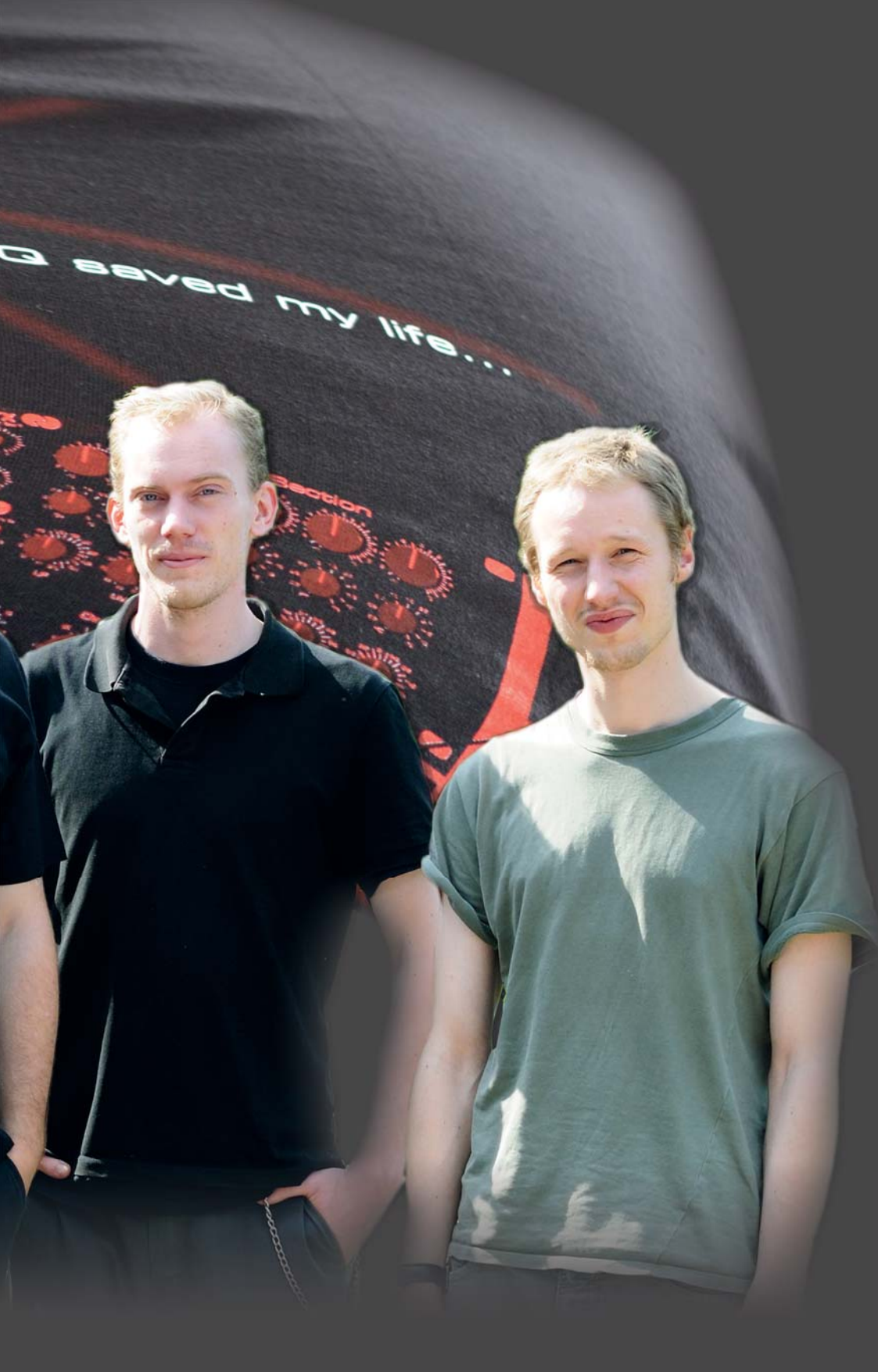
Im zweiten Teil unserer Reportage über Plug-in-Produktion erfahren Sie en detail, wie das junge Entwickler-Team von Brainworx an die Produktion von Plug-ins geht und welche Arbeitsschritte von der Idee bis zum fertigen Produkt erforderlich sind.

VON GEORG BERGER

Im ersten Teil unserer Reportage haben wir Ihnen den Software-Hersteller Brainworx vorgestellt. Sie haben erfah-

ren, welche Personen an der Produktion von Software beteiligt sind, was für Qualifikationen sie für den Job als Programmierer mitbringen, mit welchen Mitteln die Software produziert wird – C++ und

Assembler – und welches die Hauptbetätigungsfelder des Unternehmens sind. Fassen wir noch einmal zusammen: Außer der Produktion proprietärer Software-Lösungen unter dem firmeneige-



nen bx-Logo emuliert Brainworx auch Hardware. Abseits von der Schöpfung neuer Produkte treibt Brainworx mittlerweile einen nicht unerheblichen Aufwand in Sachen Service-Updates, ganz zu schweigen von den Aktivitäten im Internet, dem Hauptvertriebsweg des Unternehmens. Im zweiten Teil lenken wir jetzt den Fokus auf die eigentlichen Arbeitsabläufe bei der Plug-in-Produktion.

Im Kern geht es dabei um drei Haupt-Arbeitsschritte:

- ▶ **Schritt:** Der eigentliche Effekt-Algorithmus wird programmiert, etwa ein Kompressor
- ▶ **Schritt:** Parallel zur Entwicklung des Algorithmus wird die Bedienoberfläche, also das GUI (Graphics User Interface) entworfen, mit dessen Hilfe der Anwender die im Effekt-Algorithmus

mus bereitgestellten Funktionen und Parameter einstellen kann.

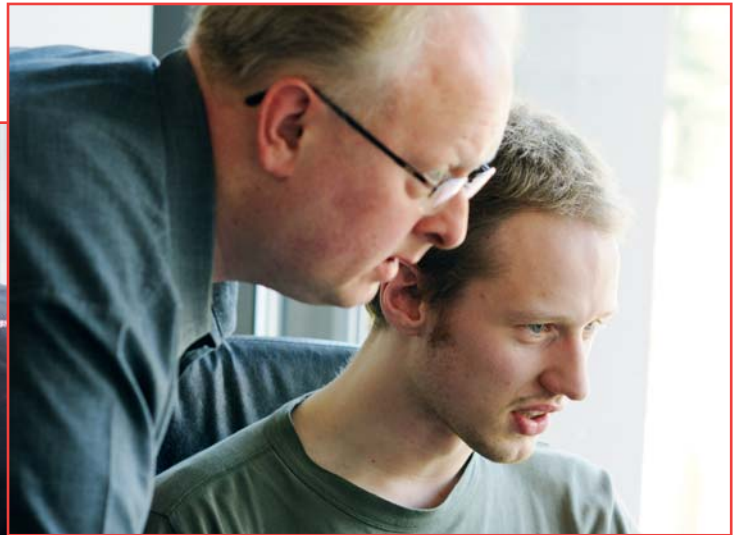
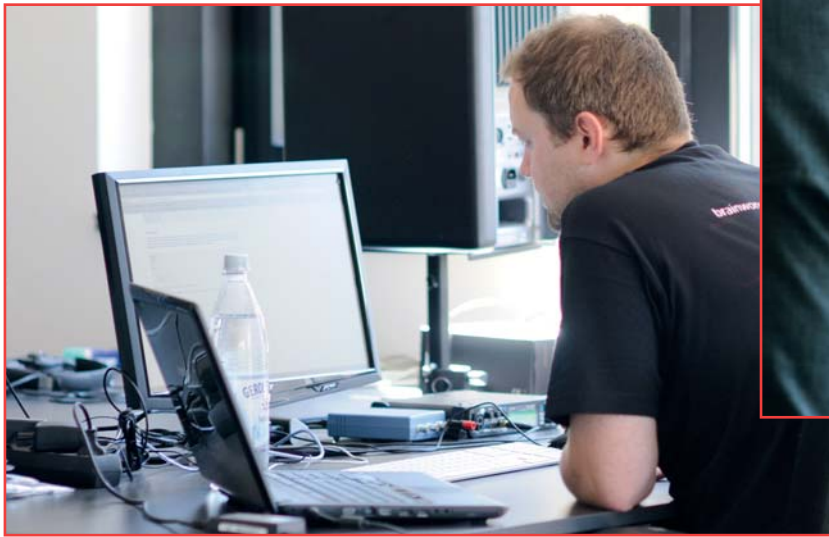
- ▶ **Schritt:** Mit Hilfe der einzelnen Schnittstellen-Formate, also VST, AU, RTAS, TDM, werden Effekt-Algorithmen und GUI schließlich „vermählt“. Anders ausgedrückt üben die Schnittstellen-Formate die Funktion eines Dolmetschers aus, der unter anderem Regler-Bewegungen im GUI übersetzt und an die relevante Funktion des Effekt-Algorithmus übermittelt.

Nicht minder wichtig ist auch das Einfügen des Kopierschutzes in die Software, was innerhalb der oben dargestellten Schritte stattfindet. Wie das genau geschieht, wollten uns die Programmierer jedoch nicht verraten, was aber auch verständlich ist.

Das Framework macht das Leben leichter

Unabhängig davon scheint der Produktions-Ablauf oberflächlich betrachtet recht banal zu sein. In der Praxis sieht das jedoch völlig anders aus, wobei gerade der dritte Schritt, also quasi die Hochzeit von GUI und Algorithmus mit einer Menge von Fallstricken durchsetzt ist. Denn jede Audio-Schnittstelle realisiert diese Übersetzungsvorgänge auf eigene Art, was entsprechend aufwändig ist. So muss beispielsweise für jede Schnittstelle der gleiche Effekt-Parameter jeweils unterschiedlich skaliert werden, um ein konstantes klangliches Ergebnis in allen Versionen zu erhalten. In der Anfangszeit von Brainworx haben die Programmierer der ersten Stunde die einzelnen Plug-in Varianten noch separat und manuell realisiert, was ein erhebliches Maß an Zeit in Anspruch nahm und wenig effizient für den Workflow war. Zudem zeigten sich mitunter Bugs in einer bestimmten Plug-in-Version, die in einer anderen jedoch nicht auftraten, was zusätzlichen Aufwand in Sachen Fehlersuche und -korrektur nach sich zog.

Wegen dieser Erfahrungen programmierte Michael Massberg bei Arbeitsbeginn im BX Team also zunächst die Ur-Version des sogenannten Frameworks, das die Algorithmen und GUIs automatisch für jedes Plug-in-Format anpasst. Besonderheit: Das Framework definiert ein eigenes Schnittstellenformat, in dem die Algorithmen programmiert werden und das sozusagen sämtliche Aspekte al-



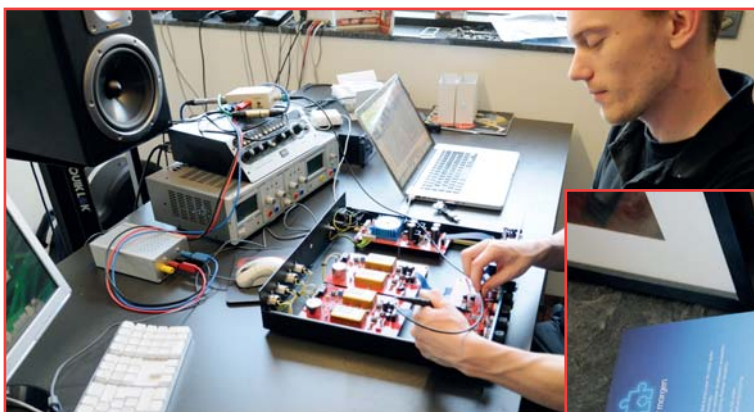
▲ Professional audio-Redakteur Georg Berger lässt sich von Reimund Dratwa die Wesenszüge der Assembler-Programmierung erläutern.

◀ André Kirchner tüftelt gerade an einem Algorithmus.

ler anderen Audio-Schnittstellen abdeckt, was von Dirk Ulrich daher treffend als „God Mode“ bezeichnet wird. Im Gespräch betont der Brainworx-Geschäftsführer, dass dieses Framework zu den Haupt-Säulen des Unternehmens zählt. Gleichzeitig vermerkt er aber auch, dass jede professionell arbeitende Software-Company über ein solches Framework verfügt, das den Programmierern beim Entwickeln neuer Software lästige Arbeiten abnimmt. Nebenbemerkung: Die Expertise von Brainworx in Sachen Framework war unter anderem einer der Gründe, die zur Kooperation mit dem amerikanischen Unternehmen Universal Audio führte. Das Brainworx-Framework zählt bei der Plug-in-Entwicklung jedenfalls zu den zentralen Werkzeugen. Das modular aufgebaute Quellcode-Konstrukt schwebt quasi über jeder Neuent-

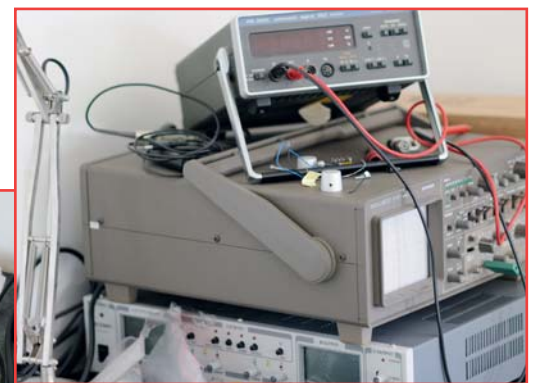
wicklung, wobei gezielt nur die für die jeweilige Schnittstelle relevanten Teile des Frameworks Einzug in die resultierende Plug-in-Version hält. Vorteil: Die Programmierer konzentrieren sich ausschließlich auf das Entwickeln des Effekt-Algorithmus und des GUI. Beide Arbeiten erfolgen dabei ein einziges Mal. Den Rest übernimmt das Framework, das beim Kompilieren, also dem Zusammenbauen des Endprodukts, am Ende gleichzeitig die VST-, AU- und RTAS-Version in 32 und 64 Bit sowohl für Windows- als auch Mac-Betriebssysteme ausgibt. Die TDM-Version ist ein Sonderfall, da der in Assembler-Code übersetzte Effekt-Algorithmus im Quelltext eingebettet ist und zur Laufzeit des Plug-ins ausgeführt wird, was mit zusätzlichen Anpassungsarbeiten zwischen GUI und Algorithmus einhergeht.

Die Pflege des Frameworks erfordert konsequenterweise ein nicht unerhebliches Maß an Aufwand, was hauptsächlich in den Zuständigkeitsbereich von Michael Massberg fällt. Sobald Updates in den SDKs der Audio-Schnittstellen, aber auch in den Sequenzern, in den Kopierschutz-Systemen und auch den Computer-Betriebssystemen erfolgen, muss das Framework blitzschnell entsprechend den Neuerungen angepasst werden. Überdies wächst das mittlerweile riesige Quell-Code-Konstrukt beständig und wird kontinuierlich um weitere Funktionen und Features erweitert. So finden sich darin etwa Funktionen zum Steuern von Parametern mit Hilfe des Mauseisens, was laut Dirk Ulrich für einige Schnittstellen über den Umweg selbst erdachter Programmier-Kniffe realisiert wird. Jüngster Funktions-Zuwachs im

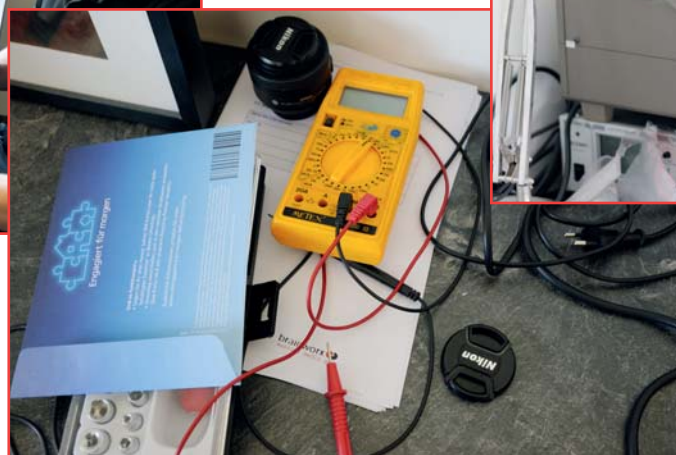


▲ Michael Massberg beim Messen der Hardware. Anschließend wird das relevante Code-Fragment solange editiert, bis die Software den gleichen Wert wie die Hardware liefert.

Jeder Arbeitsplatz ist nicht nur mit Rechnern, sondern auch mit Messgeräten und einer Abhöranlage ausgestattet. ▶



◀ Für die Messungen der Hardware werden verschiedene Geräte wie etwa Oszilloskope oder Multimeter eingesetzt.



Framework sind Über- und Unterschwinger in VU-Metern, die in Hardware zu meist beim Einschalten zu beobachten sind und künftig auch in den Plug-ins von Brainworx für zusätzliche Authentizität sorgen. Bei jedem noch so kleinen Update innerhalb des Frameworks werden sämtliche bislang produzierten Plug-ins aufs Neue kompiliert, wobei ein eigener Build-Server über Nacht läuft, um anderntags die frisch upgedateten Versionen für den Download bereitzustellen. Im Gespräch mit den Programmierern wird immer wieder deutlich, welche immense Bedeutung das Framework für die tägliche Arbeit hat. Doch auch wenn es den Programmierern eine Menge Arbeit abnimmt, sind am Ende trotzdem Kontrollen bei jedem Plug-in erforderlich, um kleine Ungenauigkeiten, die immer wieder auftreten können, zu korrigieren.

Programmieren im „God-Mode“

Befreit von vielen immer wiederkehrenden Aufgaben, können sich die Programmierer also voll und ganz dem eigentlichen Kern-Geschäft widmen, der Entwicklung von Effekt-Algorithmen und grafischen Bedienoberflächen. Letzteres ist im Vergleich zum Programmieren des Effekt-Algorithmus zumeist die leichtere Übung, wie uns von Programmierer-Seite versichert wird. Doch auch in dieser Disziplin wird teils eine Menge Aufwand getrieben, was mit Hilfe von Bildbearbeitungsprogrammen wie Adobe Photoshop realisiert wird. Bei den bx-Plug-ins, die ein einheitliches Design besitzen, kann Programmierer Stefan Schönefeld auf einen Pool an Bedienelementen zurückgreifen, die für manches neue Plug-in



In einem eigens eingerichteten (Mastering-)Studio können die Plug-ins vor Ort einem ersten Härtestest unterzogen werden.

verwendet werden können. Beim Modellieren von Hardware ist ein anderer Arbeitsablauf zur GUI-Erstellung erforderlich. Die vom Hersteller bereitgestellte Hardware wird zuerst von sämtlichen Reglern und Knöpfen befreit, so dass lediglich die nackte Frontplatte zu sehen ist. Anschließend wird sie von Michael Massberg fotografiert, wobei er auf ein kleines eingerichtetes Fotostudio nebst Profi-Blitzanlage zurückgreifen kann. Dabei wird die Frontplatte sowohl als Ganzes, wie auch in mehreren Teilaufnahmen fotografiert, in der bestimmte sinnvoll unterteilte Sektionen im Fokus stehen. Das so gewonnene Bildmaterial wird anschließend in Photoshop bearbeitet, wobei von Fall zu Fall etwa Farben korrigiert und/oder Schriftzüge und andere graphische Elemente nachgeschärft werden. Separat fotografierte Sektionen der Frontplatte werden schließlich wieder zu einem Ganzen zusammengesetzt. Zusätzlicher Aufwand wird getrieben

durch das nachträgliche Hinzufügen von Glanzlichtern und Reflexionen auf die Frontplatte, um der Bedienoberfläche einen noch realistischeren Look zu verleihen. Doch es geht noch weiter. Um Spiegelungen auf Teilen der Frontplatte oder in VU-Metern zu integrieren, hat sich Michael Massberg einen einfachen wie wirksamen Trick einfallen lassen: Dazu bedient er sich einer silbernen Kugel, von der er das sich darin spiegelnde und verzerrte Bild abfotografiert. Nicht ohne Augenzwinkern berichtet Dirk Ulrich, dass diese Spiegelungen aus dem Brainworx Tonstudio in Langenfeld stammen, das auf diese Weise ein besonderes Eigenleben in Form von Plug-ins führt. Doch zurück zur GUI-Entwicklung: Im nächsten Schritt werden die zuvor entfernten Bedienelemente – Knöpfe, Tasten, Fader, Schalter, aber auch LED-Ketten und VU-Meter – als 3D-Rendering reproduziert, wobei für die Animation der Bedienelemente eine entsprechende

Komplettieren Sie Ihre Sammlung!

Sie suchen ein früheres Heft?

Kein Problem: Alle lieferbaren Professional audio-Ausgaben, die bis einschließlich der Ausgabe 12/2009 erschienen sind, erhalten Sie für nur 1,99 Euro pro Heft zzgl. Versand bequem per Post nach Hause.

professional
audio

Das Magazin für Aufnahmetechnik

Bestellen Sie Ihre fehlenden Ausgaben auf
www.professional-audio-magazin.de/
Artikel/ältere Ausgaben



Prototyping der anderen Art: Über Pro Tools und den Einsatz diverser Plug-ins skizziert Dirk Ulrich seine Ideen für neue Plug-ins. Der Screenshot zeigt quasi die Urform des Monomakers, der etwa im Shredspread-Plug-in zum Einsatz kommt.

Zahl an Zwischenstufen erstellt wird. Für (Dreh-)Schalter muss dabei jeder Schaltzustand in Form einer eigenen Grafik vorliegen. Bei stufenlosen Drehreglern kommen schon einmal über 60 Grafiken zusammen. Mit Hilfe sogenannter Grafik-Klassen, die Teil des jeweiligen Schnittstellen-SDKs sind, lassen sich die Bedienelemente in die Frontplatten-Grafik integrieren und schließlich auch animieren. Zumeist ist eine lauffähige GUI ungleich rascher realisiert als der damit zu steuernde Effekt-Algorithmus. Je nach Art des Plug-ins, also ob es sich um ein bx-Plug-in oder ein Hardware-Modeling handelt, sind dabei unterschiedliche Herausforderungen zu meistern.

Die Königs-Disziplin: Hardware-Modeling

Im Gespräch mit den Programmierern kristallisiert sich dabei rasch heraus, dass das Modeln von Hardware zur Königsdisziplin zählt, was jedoch nicht heißt, dass ein neues bx-Plug-in „mal eben so nebenher“ entsteht. Stefan Schönefeld, der für die Produktion vieler bx-Effekte verantwortlich zeichnet, schätzt an seiner Tätigkeit die Freiheit, experimentieren zu können, was beim Hardware-Modeling nicht immer der Fall ist. Zudem bewegt er sich beim Programmieren auf rein digitaler Ebene in Sachen Audiosignal-Verarbeitung und kann dabei stets das bestmögliche, was in die-

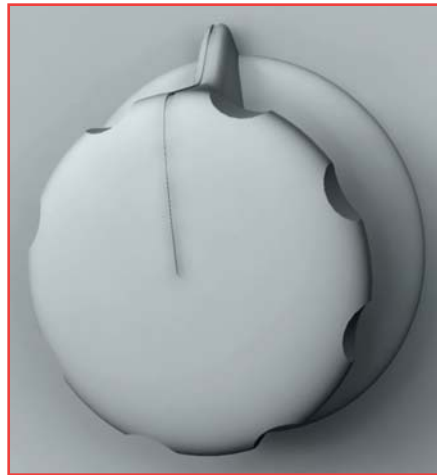
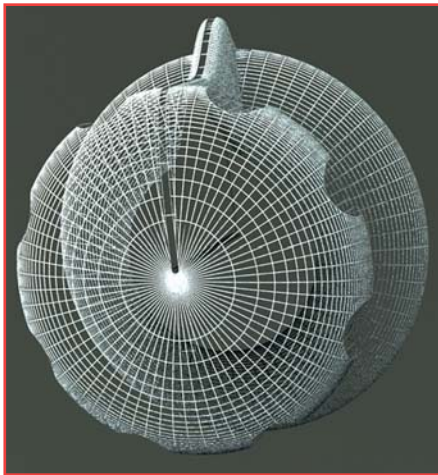
sem Bereich machbar ist, einsetzen. Bedingt durch die Art des zu produzierenden Effekts kann er teilweise auf bereits schon existierende Quellcode-Bausteine zurückgreifen, die sich rasch in den Quellcode des neuen Plug-ins einkopieren lassen. Je nach Idee, die zumeist von Dirk Ulrich stammt (siehe Screenshot oben), ist es mitunter auch möglich, ein bereits existierendes bx-Plug-in als Ausgangsbasis für die Neuproduktion zu nutzen, das um entsprechende neue Features erweitert wird. Das dabei entstandene „Frankenstein-Plug-in“ wird im weiteren Verlauf schließlich so lange weiteren Veränderungsprozeduren unterzogen, bis unterm Strich das fertige Produkt vorliegt. Wer jetzt allerdings den Eindruck bekommt, dass das Programmieren eines bx-Plug-ins dadurch eher zu den leichten Übungen gehört, was in sehr kurzer Zeit erledigt ist, der irrt. Denn je nach zu realisierender Schaltung und Funktion, gerade wenn es sich auch um völlig neue Features handelt, sitzt Stefan Schönefeld teils wochen- und monatelang an einem Projekt und tüfelt an der programmiertechnischen Umsetzung. Im Gespräch gibt er zu Protokoll, dass der dynamische Equalizer bx_dynEQ seinerzeit eine echte Herausforderung war und neun Monate Zeit (von der Idee zum fertigen Plug-in) in Anspruch nahm. Alleine für den Effekt-Algorithmus kamen dabei über 5.000 Zeilen Programmiercode zusammen, also exklusive der Programm-

teile, die noch vom Framework beige-steuert werden.

Ganz anderen Herausforderungen stehen die Programmierer beim Modeling von Hardware gegenüber. Es gilt, das Regelverhalten und die klanglichen Eigenschaften der Vorlage 1:1 auf die virtuelle Ebene zu hieven, was primär in den Aufgabenbereich von Michael Massberg und André Kirchner fällt. Zwar lassen sich beispielsweise Signalflüsse sowie Klang und Verhalten von Filtern, Dioden, Transistoren, Verstärkern und Widerständen perfekt in Quellcodes übersetzen. Doch wie so oft ist das Ganze, also das Gerät, mehr als die Summe seiner Einzelteile, was letztlich seine Magie und Einzigartigkeit in Sachen Klang und Signalbearbeitungsmöglichkeiten ausmacht. Um dies erfolgreich in Nullen und Einsen übersetzen zu können, werden zunächst die vom Hersteller übermittelten Schaltpläne studiert und analysiert. Basierend auf diesen Ergebnissen erstellen die Programmierer einen ersten Algorithmus, von Dirk Ulrich scherzhaft „das perfekte Modell“ genannt. Denn durch Revisionen in der Hardware Entwicklung können die im Schaltplan aufgezeichneten Werte und Modelle etwa für Kondensatoren und Widerstände manchmal falsch sein. Die exakten Werte kennt also oft nur der Entwickler. Bei Geräten neueren Datums ist das zwar kein Problem, doch bei der Nachbildung älteren Equipments geht der Spaß für die Programmierer oft erst richtig los. Hinzu kommen zusätzliche Unsicherheits-Faktoren wie etwa die Serienstreuung von Bauteilen, die nicht unerheblichen Einfluss auf Klang und Regelverhalten nehmen.

Mit wissenschaftlicher Akribie zum Ziel

In einem weiteren Arbeitsschritt muss daher die Hardware akribisch gemessen und die dabei ermittelten Ergebnisse mit denen des Algorithmus verglichen werden. Dazu zählt nicht nur, dass jede mögliche Reglerstellung messtechnisch erfasst und im Algorithmus angeglichen werden muss. Durch das Einlöten von Messpunkten und -brücken in die Hardware dringen die Programmierer bei dieser Tätigkeit bis auf die Bauteil-Ebene der Schaltung vor, um dort die Änderungen in der Signalverarbeitung verfolgen zu können. Bei diesem Vorgang werden Dirac-Stöße und Sinuston-Sweeps ins Ge-



Die Bedienelemente werden beim Entwickeln der Bedienoberflächen separat als 3D-Modelle gerendert. Als erstes wird dabei ein Drahtgitter-Modell des Elements erstellt, das anschließend mit entsprechenden Texturen versehen wird.

rät eingespeist und messtechnisch sowohl mit Hilfe entsprechender Mess-Hardware als auch mit Software erfasst. Michael Massberg hat sich für diesen Zweck sogar eigene Mess-Software geschrieben, die es ihm ermöglicht, die Mess-Ergebnisse von Hard- und Software parallel anzuzeigen und bequem miteinander vergleichen zu können. Liefert die Software keine identischen Werte, kann er on-the-fly solange das relevante Quellcode-Fragment anpassen, bis der Gleichstand eintritt. Solche selbst programmierten Analyse-Tools kommen übrigens auch beim Bugfixing zum Einsatz, gerade dann, wenn auftretende Fehler nicht direkt aus dem Quellcode ersichtlich sind

und das Plug-in ein auffälliges Verhalten etwa nur in bestimmten Zeitabständen zeigt. Außer der messtechnischen Angleichung des Plug-ins ans Original erfolgt selbstverständlich auch ein gehörmäßiger (akustischer) Vergleich, weshalb jeder Arbeitsplatz nicht nur mit Computern und Messgeräten, sondern auch mit einer Abhöranlage ausgestattet ist. Besonderheit: In diesem Stadium ist teilweise noch keine Audio-Schnittstelle in den Quellcode eingebunden, weshalb die Programmierer auch für diesen Fall eine entsprechende Software-Lösung selbst entwickelt haben. Alleine diese Arbeitsvorgänge – Schaltplan-Analyse, Algorithmus-Erstellung, Nachmessen der Hard-

ware plus Angleichen des Quellcodes – nehmen gehörig viel Zeit in Anspruch. Richtig knifflig wird es jedoch, wenn die Signalverarbeitung nicht linear verläuft und Messwerte etwa durch Feedback-Schleifen im Gerät stets anders ausfallen oder wenn durch Induktivitäten ein Bauteil je nach Stromstärke entsprechend stark auf andere Bauteile einwirkt. Passive Filter sind solche Kandidaten, die über das Magnetfeld ihrer Spulen auf die Spulen der benachbarten Filter einwirken. Solche komplexen Schaltungen und Wechselwirkungen sind nicht leicht digital umsetzbar, weshalb die Programmierer in dem Fall mit mathematischen Annäherungen arbeiten müssen. André Kirchner spricht in dem Zusammenhang durchaus von wissenschaftlicher Arbeit, die mit langwierigen Testphasen einhergeht. Ist auch diese Hürde genommen und das Plug-in schließlich fertig, folgt als krönender Abschluss die akustische Abnahme der Software unter professionellen Bedingungen im Brainworx-Tonstudio in Langenfeld zusammen mit dem Hersteller der Original Hardware. Dass die Programmierer in Kooperationen selbst bei kniffligen Emulationen wie dem Vitalizer und Passeq von SPL oder dem alpha compressor von Elysia exzellente Arbeit geleistet haben, davon konnten wir uns im Rahmen der einschlägigen Tests in *Professional audio* selbst überzeugen (siehe Tests in den Heften 2/2010, 1 und 2/2011). Was für ein Aufwand und Gehirnschmalz bei der Entwicklung von Audio Plug-Ins erforderlich ist, dürfte nach diesen beiden Reportage-Teilen also mehr als deutlich geworden sein. ●



Beim Hardware-Modeling wird für das GUI das Original-Gerät fotografiert und die Fotos anschließend in Adobe Photoshop aufbereitet.