Gedanken zum DUNE Prozess

P. Bastian, M. Blatt, C. Engwer, O. Ippisch, S. Lang, S. Marnach

2. Februar 2007

Vorwort

Ziel von DUNE ist es eine freie allgemeine Schnittstelle für Numerische Simulationen zu entwickeln die es der wissenschaftlichen Gemeinde ermöglicht Wissen zusammenzubringen damit weitere Forschungen darauf aufbauen können (freier Fluss des Wissens). Je mehr Entwickler sich beteiligen umso größer sind die Erfolgsaussichten und umso größer ist der Gewinn für alle Beteiligten. Dies entspricht im wesentlichen dem traditionellen wissenschaftlichen Austausch über Fachzeitschriften und Kongresse.

Bei der Entwicklung von DUNE sollen der freie Zugang (durch eine liberale Lizenz) und die Nachhaltigkeit im Vordergrund stehen. Die Arbeit muss hier auch als Investition in die Zukunft gesehen werden, kurzfristige Bedürfnisse dürfen nicht zu Lasten der Gesamtentwicklung gehen. Das Gebot der Nachhaltigkeit erfordert besondere Umsicht beim Schnittstellendesign, um die Interoperabilität und die Wartbarkeit zu gewährleisten. Dies beinhaltet auch grundlegende Designentscheidungen, wie beispielsweise die des „slim interface“.

1 Wer ist DUNE?

DUNE ist Name eines Softwareprojektes. DUNE besteht aus einzelnen, aufeinander aufbauenden Modulen. Offizielle DUNE Module werden über die Website angekündigt. Darüberhinaus kann jedermann Module anbieten, die auf DUNE aufbauen.

DUNE ist unabhängig von Einzelpersonen oder Mitgliedern.

2 DUNE Modulstatus

2.1 Kernmodule

Kernmodule sind Module, welche als weitgehend stabil angesehen werden und die aufgrund Ihrer zentralen Rolle eine besondere Bedeutung haben. Sie sind Bestandteil von DUNE Releases (siehe 6) und Änderungen sind hier mit großer Sorgfalt durchzuführen.

Zur Zeit wären dies `dune-common`, `dune-grid` und `dune-istl`, sowie `dune-grid-howto`.

2.2 Infrastrukturmodule

Infrastrukturmodule bieten Systeme, welche der Entwicklung von DUNE helfen, selbst aber nicht direkt Bestandteil des Frameworks sind. Hierzu zählen beispielsweise `dune-web` und `dune-autobuild`. Diese Module müssen auf die Kernmodule abgestimmt sein, sind aber nicht Bestandteil eines DUNE Releases.

2.3 Experimentelle Module

Während der Entwicklung eines neuen Moduls ist es nötig die Regeln sehr offen zu gestalten. Man braucht Spielraum für Experimente und auch für Fehler. Bevor ein Modul stabil genug ist, um ein Kern Modul zu werden ist es ein experimentelles Modul, bei dem die in den folgenden Sektionen beschriebenen strengen Regeln keine Anwendung finden. Trotzdem ist die DUNE Website für diese Module immer noch die zentrale Anlaufstelle.

2.4 Externe Module

Durch seine Modularität bietet DUNE auch externen Entwicklern die Möglichkeit eigene Module zu entwickeln. Diese Module sind daher grundsätzlich keine Kernmodule. Sie werden unabhängig von der eigentlichen DUNE Entwicklung gepflegt, können aber auf der Website verlinkt werden.

3 DUNE Mitgliederstatus

In der Regel werden die Module unabhängig von einander entwickelt.

Es gibt die folgenden Status von an DUNE mitwirkenden Personen, diese beziehen sich immer auf ein Modul. Diese Einteilung bezieht sich primär auf die Kernmodule.

Anwender Jemand der DUNE benutzt. Alle DUNE Kernmodule sind öffentlich zugänglich.

Modul-Entwickler Jemand der Schreibrechte auf das Repository besitzt. *Die Vergabe dieses Status sollte relativ restriktiv gehandhabt werden*

DUNE-Entwickler Jeder Entwickler mit Entwickler-Status bei einem Kernmodul ist auch ein DUNE-Entwickler. *(Alternative: Jeder Entwickler, der Modul-Entwickler eines Moduls ist, das auf `dune-project.org` gehostet wird)*

Der Mitgliederstatus bezieht sich nur auf Einzelpersonen. Weitere Hierarchieebenen gibt es nicht.

3.1 Aufnahmeprozess für DUNE-Entwickler

Um Entwickler-Status für ein DUNE Modul zu erhalten, muss man den folgenden Aufnahmeprozess durchlaufen.

1. **Empfehlung durch einen anderen Entwickler.**

   Der Befürworter hat sich von den Fähigkeiten des Bewerbers überzeugt. Er unterstützt die Aufnahme und vertritt diese Position gegenüber den anderen Entwicklern (dies geschieht in einer Art Laudatio).

   Die Empfehlung wird üblicherweise aufgrund guter Codebeiträge zu einem der DUNE Module (nicht notwendigerweise das, für welches der Antrag läuft) ausgesprochen. Der Vorgeschlagene sollte also schon einige Beiträge zu dem Modul gemacht haben, die von seinem Laudator oder anderen DUNE Developern eingecheckt wurden.

2. **Probezeit**

   Nach der Empfehlung wird der Bewerber für 4 Wochen zum Entwickler auf Probe. Während dieser Zeit hat er bereits Schreibzugriff auf das Repository, allerdings noch kein Stimmrecht.

3. **Abstimmung**

   Die Mitarbeiter mit vollem Entwickler-Status stimmen gemäß des üblichen DUNE-Abstimmungsmodus (siehe 5.3) über die Aufnahme ab.

4. **Voller Entwickler-Status**

   Von nun an hat der neue Entwickler den vollen DUNE-Modul-Entwickler-Status, mit vollem Stimmrecht.

3.2 Ruhenlassen der Mitgliedschaft

DUNE-Entwickler können freiwillig für eine gewisse Zeit ihre aktive Mitgliedschaft ruhen lassen. Dies ist hilfreich, wenn man z. B. während eines längeren Forschungsaufenhaltes keine Zeit für DUNE haben wird, aber dennoch nach der Rückkehr weiter entwickeln möchte.

Während die Mitgliedschaft ruht, verzichtet der Entwickler auf sein Stimmrecht. Dadurch wird verhindert dass das Quorum bei Abstimmungen während der Abwesenheit unnötig erschwert wird.

4 Einbringen von Änderungen

Die hier beschriebenen Regeln gelten nur für echte Änderungen, Bugfixes sind davon ausgenommen.

Ein Bugfix zeichnet sich dadurch aus, dass er

- einen Fehler behebt,
- das Interface nicht ändert (d.h. auch nicht erweitert),
- die Semantik nicht ändert oder nur einen beschlossenen und dokumentierten Zustand wieder herstellt.

Änderungen können klein (ein Parameter in einer Methode) oder groß sein. Da Größe und Auswirkungen von Änderungen stark schwanken ist die Verhältnismäßigkeit zu wahren.

Eine Änderung ist grundsätzlich als Vorschlag an die Mailingliste des Moduls zu richten. Folgende Dinge sollten enthalten sein:

- Beschreibung wozu das gut ist.
- Wie soll die Änderungen realisiert werden. Eventuell sind mehrere Alternativen vorzustellen.
- Auswirkungen auf den restlichen Code sind darzulegen.
- Wer soll das machen bzw. ist involviert.
- Zeithorizont.
- Wichtigkeit der Änderung.
- Liegt schon eine Pilotimplementierung vor, welche Erfahrungen gibt es damit?

Wir stellen uns vor, dass es da ein Template gibt an das man sich im wesentlichen hält. Diese Dokumente sollten von einem System verwaltet werden damit der Diskussionsprozess nachvollzogen werden kann.

5 Entscheidungsfindung

- Zunächst sollten Änderungsvorschläge über einen angemessenen Zeitraum über die Mailingliste oder ein anderes System diskutiert werden. Hierbei sollte informell abgeklärt werden, ob alle diese Änderung gut finden bzw. wie sie sich das anders vorstellen würden. So kann ein Änderungsvorschlag bereits vor der eigentlichen Implementierung iterative durch das Feedback der developer *und* user verbessert werden. Die Vorschlagenden können anhand der Diskussion auch abschätzen, ob die zu erwartende Akzeptanz den Aufwand einer Prototyp Implementierung rechtfertigt.
- Die Vorschlagenden und/oder andere implementieren, den Änderungsvorschlag in einem eigenen Branch. Auch konkurrierende Implementierungen sind möglich.
- Sobald eine Prototyp Implementierung vorhanden ist, kann die eigentliche Diskussion sowie Abstimmung erfolgen. Hierbei dient der Prototyp vor allem zur Überprüfung der Auswirkungen auf andere Teile von Dune sowie Applikationen. Die Kriterien sind dem Abstimmungsmodus, Abschnitt 5.3, zu entnehmen. In dieser Abstimmung kann auch die Code-Qualität geprüft bzw. verbessert werden.

- In der Regel ist Einvernehmen unter den Entwicklern des betroffenen Moduls zu erreichen. Falls eine Abstimmung notwendig wird, erfolgt diese nach dem DUNE-Abstimmungsmodus; das ist auch für Kernmodule sinnvoll, da Änderungen in diesen Modulen in der Regel alle darauf aufbauenden Module und Modelle betreffen, so dass diese Änderungen restriktiv gehandhabt werden sollten.
- Es sollte einen Satz von Grundregeln („Verfassung“) geben, der unantastbar ist. Zum Beispiel
  - „slim interface“
  - „Softwarequalität ist wichtiger als die Geschwindigkeit der Änderung“
  - ...
- Jeder Entwickler spricht für sich selbst.

5.1 Entscheidungen, die einzelne Kernmodule betreffen

Dies betrifft hauptsächlich die Schnittstelle, bzw. Entscheidungen, welche Features in die nächste Release aufgenommen werden und welche nicht. Hierbei sind die Entwickler des betreffenden Moduls zu fragen.

5.2 Entscheidungen, die das Gesamtprojekt betreffen

Fragen in diesem Bereich sind beispielsweise die Aufnahme neuer Module als Kernmodule, Schnittstellen zwischen Modulen, DUNE-Releases, etc. Solche Entscheidungen sollten von allen DUNE-Entwicklern, bzw. von denen der betroffenen Module gefällt werden.

5.3 Abstimmungsmodus

Sofern nicht anders beschrieben, finden Abstimmungen in DUNE nach folgendem Modus statt.

1. Bei Modulabstimmung ist jeder aktive Modul-Entwickler stimmberechtigt, bei Entscheidungen die das Gesamtprojekt betreffen alle Entwickler der Kernmodule. Jeder Abstimmungsberechtigte hat eine Stimme.
2. Die Abstimmung erfolgt während eines Zeitraums von x Tagen (wir schlagen 4 Wochen vor).
3. Die Abstimmung ist ungültig, wenn weniger als 50 % der Stimmberechtigten abgestimmt haben.
4. Ein Antrag ist angenommen, wenn mindestens 80 % der abgegebenen Stimmen dafür waren.

6 Releasemanagement

Bisher gibt es keinen konkreten Vorschlag, aber in Zukunft muss man die Release besser organisieren.

Man sollte den Ansatz „Release early, release often“ verfolgen, um neue Entwicklungen Anwendern zum Testen anzubieten, ohne sie gleich zur Verwendung der Entwicklungsversion zu zwingen. Hierbei könnte man auch nicht so gut abgestimmte „Technology Previews“ in Erwägung ziehen.

Bei der letzten Release hat es sich als überaus schwierig erwiesen, den Featurefreeze wirklich durchzusetzen. Man sollte daher früher und öfter branchen und den Umfang der Änderungen klein zu halten.

Wenn in einem der Module grundlegende Umstrukturierungen im Interface nötig sind, sollte das durch einen entsprechenden Versionssprung gekennzeichnet werden.

Änderungen im Interface sollten nach Möglichkeit so eingebaut werden, dass der alte Code mindestens noch eine Release lauffähig ist und deprectated Warnungen erhält.

Es ist wünschenwert, dass es einen Verantwortlichen für eine Release gibt, der sich um deren Organisation kümmert. Er ist hierbei ausführendes Organ, nicht Entscheidungsträger.

7 Lizenz, Kommerzialisierung

- Da DUNE bereits unter der LGPL veröffentlicht wurde, kann die Lizenz nur noch gemeinschaftlich, d.h. nach Zustimmung aller Beteiligten, geändert werden.
- Ein Problem mit der jetzigen Lizenz ist, dass die LGPL das „linken“ einer *Template* Bibliothek in einem nicht freien Projekt nicht erlaubt, da es sich hierbei strenggenommen nicht um „linken“ handelt, sondern um kompilieren.
- Ein weiterer Punkt ist, dass wir auch die Möglichkeit behalten möchten, Bibliotheken unter der GPL zu verwenden – wie beispielsweise ALUGrid und Alberta. Das erfordert, dass auch wir eine freie Lizenz verwenden.
- Niko hat einen Punkt: Jede Entscheidung hier muss auch offiziell mit den Geldgebern, mindestens der Universität abgesprochen sein. Hier ist äusserste Vorsicht ratsam. Wenn die Universität auch nur ahnt, dass hier Geld zu machen wäre ist es aus mit open source.
- Benedikt hat einen Punkt: Es ist wahrscheinlich ziemlich schwer eine Geldabgabe einzufordern: Wieviel wird in welchem Fall fällig? Wer nimmt das Geld? Ein DUNE-Verein? Dieser kann sicher nicht ohne Kenntnis der Geldgeber exisitieren. Diese möchten dann eventuell mehr Geld verlangen und das Geld selbst einstecken? Wer bestimmt dann wie das Geld ausgegeben wird?

Aufgrund der vielen Schwierigkeiten sehen wir eigentlich nur die beiden folgenden Möglichkeiten:

1. Es bleibt so wie es ist, als LGPL. Konsequenzen:

   - Jemand kann auf DUNE aufbauen und für das Resultat Geld verlangen. Verändert er dazu DUNE, bzw. nimmt er Code von DUNE in seinen Aufsatz hinein so muss er diesen veröffentlichen. Linkt er nur gegen DUNE so muss er seinen Teil nicht veröffentlichen.
   - Wir können also nicht verhindern, dass jemand mit DUNE Geld verdient. Wem das nicht passt, der kann es ja auch machen.
   - Wenn jemand mit DUNE Geld verdient wird er an einer langfristigen stabilen Entwicklung von DUNE interessiert sein und DUNE fördern. Implizit kommt das also dem Projekt zugute.

2. Wir verwenden die GPL mit „linking exception“. Siehe hierzu auch

   - `http://mailman.fsfeurope.org/pipermail/discussion/2006-October/006386.html`
   - `http://mailman.fsfeurope.org/pipermail/discussion/2006-October/006420.html`
   - `http://gcc.gnu.org/onlinedocs/libstdc++/17_intro/license.html`

   Es geht dabei um ein spezielle Ausnahme, die das komplette Verwenden der Bibliothek, d. h. sowohl Linken als auch Verwenden von Templates und Inlining ohne Einschränkung erlaubt.

   Es wird empfohlen, das nicht als Ergänzung zur LGPL sondern zur GPL zu machen. Die LGPL ist eine komplett eigene Lizenz. Bei Version 3 überlegt man jetzt, die LGPL als GPL + Ausnahmen zu regeln, aber bisher ist es eine eigene Lizenz. Die GPL ist in vielen Belangen klarer formuliert, deswegen wird empfohlen GPL + linking exception zu verwenden.

3. Wir verbieten nicht freie Aufsätze generell, indem wir zur GPL wechseln. Geld kann dann im Prinzip nur mit Distribution und Support verdient werden. Eventuell vergrault man damit eine Reihe von potentiellen Benutzern.

An die Unis/Geldgeber sollten wir mit dem Wunsch herantreten, uns das Veröffentlichen des Quellcodes unter der (L)GPL zu erlauben. Als Begründung könnte man anführen, dass DUNE ein reines Grundlagenforschungsprojekt ist, mit dem in dieser Form kein Geld verdient werden kann. Es entspricht also der Veröffentlichung eines mathematischen Satzes oder eines Algorithmus – den kann auch jeder kostenlos verwenden. Ziel ist es, der Wissenschaftlichen Gemeinde Wissen zur Verfügung zu stellen, damit diese weitere Forschungen darauf aufbauen kann (freier Fluss des Wissens). Gott sei Dank gibt es hier noch keine Softwarepatente.

Es sollte über regelmäßige Veröffentlichungen gesichert werden, dass die Verwendung von DUNE angemessen zitiert werden kann und die beteiligten Entwickler dadurch angemessen gewürdigt werden.