

Alisia – jetzt seetüchtig

Wie aus einem in die Jahre gekommenen Mailinglisten-Manager ein modernes, DSGVO-konformes Modul in einer regelbasierten Event-Architektur wurde.

von **Michael Baas** · Dyalog / APL Germany

Alisia ist der Mailinglisten-Manager von APL Germany. Was als überschaubare Modernisierung gedacht war, entwickelte sich zu einer Reise mit einigen Eisbergen – vom Umgang mit Umlauten über die Eigenheiten von Outlook-Thread-Formatierung bis hin zu Spamfiltern. Dieser Artikel fasst die wichtigsten Stationen zusammen, beschreibt die neue Architektur *Konducta*, in die Alisia nun eingebettet ist, und berichtet offen vom Einsatz von KI-Werkzeugen in der Entwicklung.

1. Warum die Reise überhaupt?

Alisia ist seit vielen Jahren das Rückgrat der Kommunikation in der APL-Germany-Community. Sie verwaltete die Liste der Abonentinnen und Abonenten, verteilte eingehende Nachrichten und sorgte dafür, dass ein- und ausgehende Mails konsistent aussahen. So weit, so gut – doch in ihrer ursprünglichen Form war sie ein Kind ihrer Zeit: reine Plain-Text-Mails, kein Bounce-Handling, kein MIME, keine DSGVO-Funktionen, nur eine Mailingliste und ausschließlich lauffähig unter Windows.

Die Aufgabe schien klar: modernisieren, aufräumen, HTML- und MIME-Unterstützung ergänzen, auf Linux portieren und ein paar Features nachziehen. Das Schiff sollte in ruhigeres Fahrwasser geführt werden. Was zunächst nach einer überschaubaren Refaktorisierung aussah, entpuppte sich als Fahrt durch ein Feld von Eisbergen. Keiner davon war existenzbedrohend – aber jeder hat mehr Zeit gekostet als geplant.

2. Drei Eisberge

Im Nachhinein lassen sich die drei Kernprobleme gut benennen. Sie teilen ein gemeinsames Muster: Sie sind nicht in der Spezifikation zu finden, sondern entstehen aus dem Zusammenspiel mit der realen Welt – mit Mail-Clients, Transportwegen und Spamfiltern.

2.1 Umlaute

Der erste Eisberg zeigte sich in einer harmlos wirkenden Test-Mail. Subject: »ALISIA #241 – fröhliche umlaut-tests: ÄÖÜäöüáéóúÀÈÌÒÙàèìòùâêîôûß«. Während Apple Mail und Thunderbird das sauber zurückschickten, machte Outlook daraus eine Fülle Fragezeichen. Auch bei der Behandlung von Umlauten im Mailtext gab es Erfolge – und Misserfolge, sobald andere Mailclients beteiligt waren.

Die Ursache saß tiefer als gedacht: Alisia stützt sich auf MimeKit, um Mails zu erzeugen und zu parsen. An verschiedenen Stellen wurden aber Aufrufe ohne explizite UTF-8-Codierung verwendet – mit dem Ergebnis, dass MimeKit auf plattformabhängige Default-Kodierungen zurückfiel. Die Lösung war unspektakulär, aber schwierig zu finden: Der Parameter »utf-8« wurde konsequent überall dort

ergänzt, wo MimeKit eine Kodierung entgegennimmt. Damit verschwanden sowohl die kaputten Betreffzeilen als auch die unterschiedlichen Darstellungen zwischen Outlook und Android.

Interessant an diesem Eisberg: Die eigentliche *Diagnose* – also das systematische Einkreisen, welche Stellen des Codes die Kodierung verlieren – wurde massiv durch den Einsatz einer KI beschleunigt. Der *Fix* war trivial; der Weg dorthin nicht.

2.2 Mail-Threads und Outlook

Der zweite Eisberg hatte nichts mit falscher Codierung zu tun, sondern mit Ästhetik – und mit einem Mail-Client, der sich standhaft weigert, HTML so zu lassen, wie es ist. Die Idee: In langen Threads sollte die Hintergrundfarbe von Kopf- und Fußzeilen mit zunehmender Tiefe abgeschwächt werden, damit man auf einen Blick erkennt, wie weit in die Vergangenheit ein Zitat zurückreicht. In der Praxis also ein klassisches Nested-Quoting, nur visuell sauber aufbereitet.

Outlook verändert das HTML eingehender Mails jedoch erheblich: Inline-CSS wird umgeschrieben, Klassen werden ersetzt, Tabellen werden neu strukturiert. Wer versucht, seine eigenen Header anhand einer Klasse oder ID wiederzufinden, stößt schnell an Grenzen. Die Lösung, die aktuell produktiv ist, setzt stattdessen auf robuste Merkmale: eine Kombination aus Font, Farbe und Position im DOM, die auch nach Outlooks Transformationen noch identifizierbar ist. Zusätzlich erkennt Alisia typische Transformationen verschiedener Mail-Clients und korrigiert diese, bevor der eigene Algorithmus greift.

2.3 Spamfilter

Der dritte Eisberg war der unsichtbarste. Ausgehende Mails wurden gelegentlich von Spamfiltern geblockt oder in den Junk-Ordner verschoben – ohne dass es eine einzelne offensichtliche Ursache gegeben hätte. Im Laufe der Analyse zeigte sich ein ganzes Bündel kleiner Themen: ein Footer, der deutlich als Listen-Footer erkennbar sein muss, funktionierende Unsubscribe-Links, ein klar formatierter Listen-Header, passende DKIM- und SPF-Einträge auf Serverseite sowie eine Datenschutz-Verlinkung, die Filter zunehmend erwarten.

Alisia sendet inzwischen einen standardisierten Footer mit Listenname, Versionsinformation, Unsubscribe-Hinweis und einem Link zur Datenschutzerklärung.

3. Neue Features

Parallel zum Lösen der Eisberge wuchs das Feature-Set spürbar. Die folgenden Punkte sind die aus Anwendersicht relevantesten Neuerungen:

- **NOMAIL / MAIL:** Pause auf Abruf. Wer für ein paar Tage keine Listen-Mails möchte, schickt eine Mail mit Betreff »nomail« und schaltet sich mit »mail« wieder scharf. Das ist besonders in Urlaubszeiten ein häufig gewünschtes Feature gewesen.
- **Alternative Absenderadressen:** Über GETMYDATA und SETMYDATA können Nutzer einsehen, welche Daten über sie gespeichert sind, und mehrere Absenderadressen hinterlegen. Damit

funktioniert die Liste auch, wenn jemand mal vom Smartphone, mal vom Büro-Account aus schreibt. (Hierzu gab es Diskussionen über Erweiterungen, die auch

- **Bounce-Erkennung:** Für jeden Empfänger führt Alisia einen Bounce-Zähler. Bei wiederholt nicht zustellbaren Nachrichten wird der Eintrag automatisch deaktiviert und die Administration informiert. Tote Einträge staubten früher still in der Liste – jetzt fallen sie gezielt auf.
- **DSGVO-Funktionen:** Datenschutzerklärung, maschinelle Einsicht in die eigenen Daten (GETMYDATA) und Löschung sind jetzt automatisierte Feature.
- **Mehrsprachigkeit:** Alle Systemtexte und automatischen Antwortmails lassen sich lokalisieren. Aktuell ausgeliefert werden Deutsch und Englisch; weitere Sprachen können leicht bedient werden.
- **Cross-Platform:** Entwickelt unter Windows, läuft Alisia inzwischen ohne gesonderten Portierungs-Aufwand auch unter Linux (Dyalog v20). Damit eröffnen sich neue Deployment-Optionen, insbesondere auf günstigen VPS-Instanzen.

3.1 Steuerung über den Betreff

Alisia wird – bewusst – nicht über ein Web-Interface, sondern über den Mail-Betreff gesteuert. Das mag altmodisch wirken, hat aber einen entscheidenden Vorteil: Man braucht keinen zusätzlichen Account, kein Passwort, keine zweite App. Jede Nutzerin und jeder Nutzer kennt den eigenen Mail-Client. Die folgenden Kommandos stehen zur Verfügung:

Betreff	Wirkung
[un]subscribe	An-/Abmeldung am Verteiler
help	Zusendung der Dokumentation
nomail	Pause auf Abruf – etwa während des Urlaubs
mail	Pause beenden, wieder aktiv im Verteiler
GETMYDATA	Einsicht in die gespeicherten Daten (DSGVO)
SETMYDATA	Alternative Absenderadressen hinterlegen

Groß- und Kleinschreibung ist irrelevant. Eine deutsche Übersetzung der Befehle wäre möglich, der Preis wären allerdings zwei parallele Vokabulare; bleiben daher bei den englischen Begriffen.

4. Konducta: Alisia als Modul

Der vielleicht größte konzeptionelle Schritt war nicht in Alisia selbst zu finden, sondern darum herum. Alisia ist in der neuen Variante kein monolithischer Server mehr, sondern ein **Handler** in einem regelbasierten Event-System mit dem Arbeitsnamen *Konducta*.

Das Grundmuster ist einfach. Ein Event – zum Beispiel »neue Mail in alisia@apl-germany.de« – wird an Konducta übergeben. Konducta prüft die konfigurierten Regeln und leitet das Event an den zuständigen Handler weiter. Für Mailinglisten-Verkehr ist das der Alisia-Handler; für Benachrichtigungen über neue GitHub-Releases ein separater Handler.

Für Alisia hat diese Entkopplung mehrere angenehme Konsequenzen. Erstens: Es lassen sich problemlos mehrere Mailinglisten betreiben, einfach indem mehrere Alisia-Handler konfiguriert werden. Zweitens: Der gesamte Infrastruktur-Kram (Mail abholen, Events erzeugen, auf Fehler reagieren) ist nicht mehr im Code der Mailingliste selbst vergraben, sondern an einer Stelle sichtbar.

5. Nächster Hafen: Performance

In dieser Phase stand Performance bewusst nicht im Vordergrund. Funktionalität und Stabilität hatten Priorität, und im Betrieb mit rund 100 Abonnenten ist die Mengensituation entspannt. Trotzdem lohnt es sich, den nächsten Kurs jetzt zu skizzieren, bevor die Zahlen weiter wachsen.

- **Profiling zuerst:** Bevor optimiert wird, muss klar sein, wo Zeit verloren geht. Eine erste Hypothese ist, dass ein großer Teil der Laufzeit im Logging steckt – ein vergleichsweise günstiger Hebel, den man durch geschicktes Log-Level-Management weitgehend entschärfen kann.
- **Mail-Verarbeitung parallelisieren:** Derzeit werden Mails sequenziell verarbeitet. Mit ein paar Dyalog-Threads lässt sich hier ohne großen Aufwand spürbar beschleunigen.
- **IMAP-Idle statt Polling:** Alisia fragt aktuell in festen Intervallen den Mail-Server ab. Ein Umstieg auf IMAP-Idle würde Latenz deutlich reduzieren und Server-Last sparen.
- **Caching:** Für häufig gelesene Operationen – etwa das Ermitteln aktiver Empfänger einer Liste – lohnt sich ein kleiner Cache. Auch hier gilt: erst messen, dann cachen.

Kurz gesagt: erst das Logging reduzieren, dann messen – und wahrscheinlich ist die Lösung danach bereits gut genug für 100 und mehr Abonnenten. (Das waren übrigens Vorschläge der KI – und ich gebe zu, paranoid geloggt zu haben, insofern ist dieser Vorschlag schon berechtigt. Die übrigen Punkte allerdings halte ich nicht für erfolversprechend - so „groß“ ist das Problem nicht)

6. KI als Entwicklungswerkzeug

Ein Wort zu einem Thema, das aus einem Projekt dieser Art kaum mehr herauszurechnen ist: dem Einsatz von KI als Werkzeug. Der Punkt ist weniger, ob man KI nutzt, sondern wofür und mit welchem kritischen Filter.

- **Regex-Entwicklung:** Für das Parsen von Mail-Headern, die Normalisierung von Betreffzeilen und das Erkennen typischer Thread-Marker wurden reguläre Ausdrücke KI-gestützt erstellt und getestet. Die Produktivität hier ist enorm, sofern man jedes Ergebnis gegen eigene Testfälle laufen lässt.
- **Edge-Case-Tests:** Bei Themen wie Threading, Sonderzeichen-Kodierung und MIME-Grenzfällen hat die KI geholfen, Szenarien systematisch durchzuspielen, die einem im Alltag vermutlich nicht von selbst eingefallen wären.
- **Flag-Recherche:** Bei MimeKit ließ die KI mehrere Optionen finden und bewerten, die in der bisherigen Implementierung schlicht übersehen worden waren. Die eigentliche Korrektur war dann wieder nur eine Zeile Code, doch ohne die Vorarbeit hätte sie länger gebraucht.

- **Präsentation:** Auch der Entwurf der begleitenden Präsentation wie auch dieses Artikels – inklusive eines Großteils der verwendeten Grafiken – entstand auf Basis unsortierter Stichpunkte in enger Zusammenarbeit mit einer KI.

Das Muster ist in allen vier Fällen das gleiche: Die KI beschleunigt *Diagnose* und *Vorarbeit* dramatisch, während Entwurfsentscheidungen, Tests und das Lesen des produktiven Codes weiter in der Verantwortung des Menschen liegen. Wer das umdreht, baut sich Probleme ein, die erst Monate später auffallen.

7. Fazit: Das Schiff ist seetüchtig

Aus einer »kleinen Modernisierung« ist ein Projekt mit echten Zähnen geworden. Die Kernprobleme rund um Kodierung, Thread-Darstellung und Zustellbarkeit sind gelöst, das Feature-Set ist substanziell gewachsen, DSGVO-Anforderungen sind sauber abgedeckt und die Code-Basis ist deutlich kompakter als vorher. Durch die Einbettung in Konducta ist Alisia außerdem keine Insel mehr, sondern ein Modul unter mehreren – beliebig viele Mailinglisten und beliebige weitere Event-Handler sind einfach eine Konfigurationsfrage.

Wir hoffen nun, durch Erweiterung des Teilnehmerkreises (gerne mit „exotischen“ Mailclients!) neue Testfälle zu entdecken und die Einsatzfähigkeit beweisen zu können.

Für heute aber gilt: Das Schiff ist wieder auf Kurs. Die See ist ruhiger geworden. Der Blick richtet sich auf den nächsten Hafen – nicht mehr auf die Eisberge im Rücken. ⚓

Kontakt

Michael Baas · mbaas@dyalog.com

Mailingliste: alisia@apl-germany.de

Source code: <https://github.com/mbaas2/Konducta-Alisia> und <https://github.com/mbaas2/Konducta>