

Schluss mit "one size fits all" – modellbasiertes Variantenmanagement

Teil 3: Nutzung eines spezialisierten Werkzeuges für Variantenmanagement

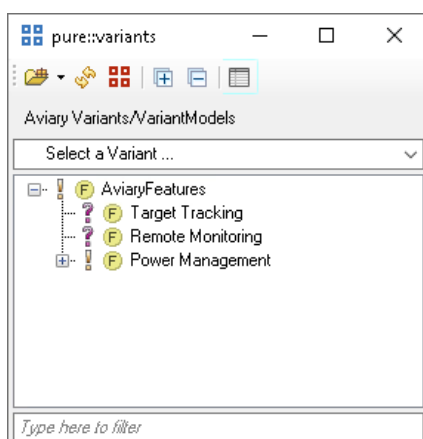
Eine verbreitete Herangehensweise ist die Abbildung des gewünschten Spektrums von Varianten in Form eines „**Feature Tree**“ Modells, das als dediziertes Metamodell ausschließlich der Beschreibung der Varianten dient und von allen Projektbeteiligten genutzt wird. In der praktischen Entwicklungsarbeit wird dieses Feature Tree Modell in einem dedizierten Werkzeug wie pure::variants angelegt und verwaltet. Die Verknüpfung von pure::variants mit den im Projekt verwendeten Engineering Tools ermöglicht allen Projektbeteiligten den direkten Zugriff auf die aktuelle und vollständige Dokumentation der Varianten und verhindert das Divergieren von Planung und Realisierung.

Wie der Ablauf im Projekt konkret aussehen kann, zeigen wir in einer integrierten Umgebung mit den IBM Engineering Tools DOORS Next, Rhapsody und Engineering Workflow Management (EWM) mit dem auf Variantenmanagement spezialisierten Werkzeug pure::variants.

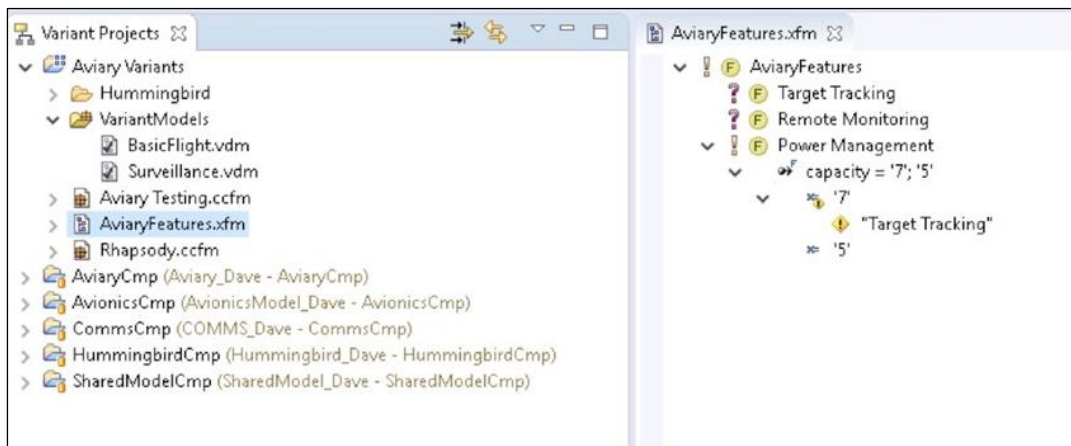
Das Kolibri Beispiel – Definition von Varianten und Features

In unserem Beispiel betrachten wir eine Drohne (im Modell als „Hummingbird“, deutsch Kolibri, bezeichnet) mit den Features „Target Tracking“, „Remote Monitoring“ und „Power Management“. Dabei sind „Target Tracking“ und „Remote Monitoring“ optionale Features. „Power Management“ ist eine Basisfunktion, die in allen Varianten vorhanden sein muss, da der Betrieb der Drohne ohne Spannungsversorgung nicht möglich ist. Die Auslegung der Spannungsversorgung für eine Variante ist abhängig von den in dieser Variante gewählten optionalen Features: durch den erhöhten Strombedarf der Funktion „Target Tracking“, muss bei Auswahl dieser Funktion die Spannungsversorgung mit 7Ah parametrisiert werden, ansonsten sind 5Ah ausreichend.

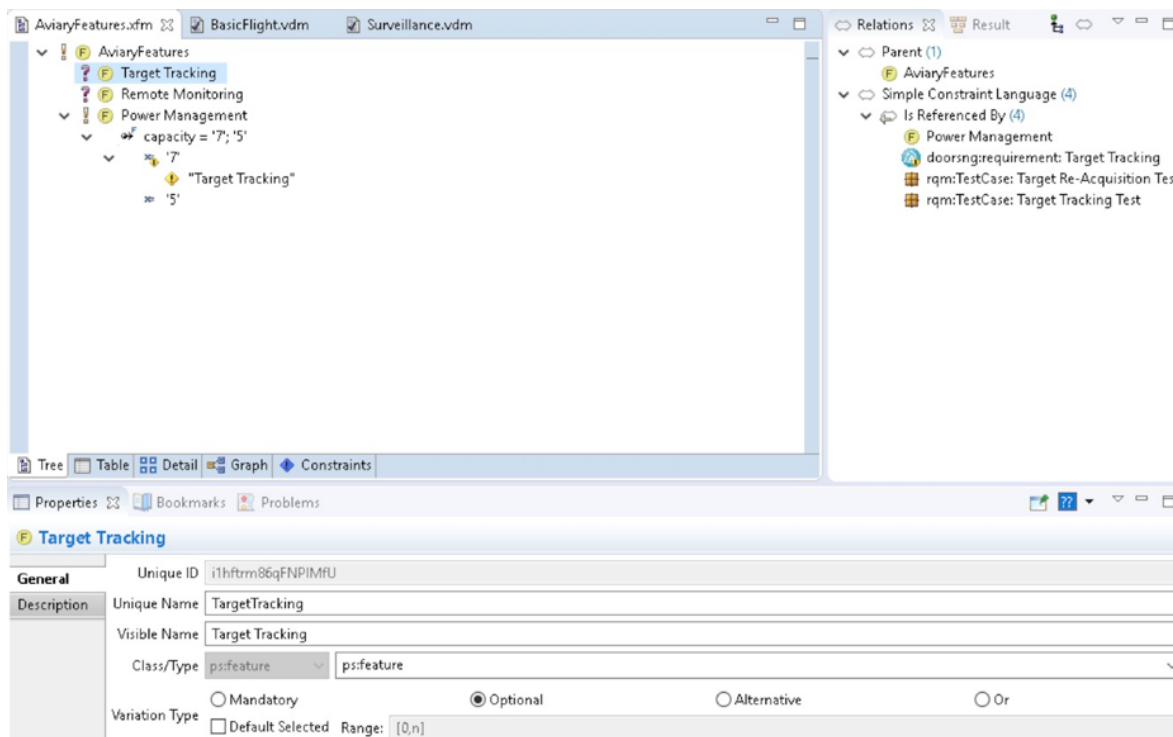
In pure::variants wird das vollständige Feature-Modell zentral verwaltet und kann von unterschiedlichen Tools entlang des Lifecycle genutzt und bei Bedarf bearbeitet werden. Die Darstellung in pure::variants zeigt dieser Screenshot:



Aus der Sicht von IBM Engineering Workflow Management wird das vollständige Feature-Modell dargestellt, exakt so, wie es in der pure::variants GUI erscheint, einschließlich aller verpflichtenden und optionalen Features (im Screenshot durch das Fragezeichen markiert):

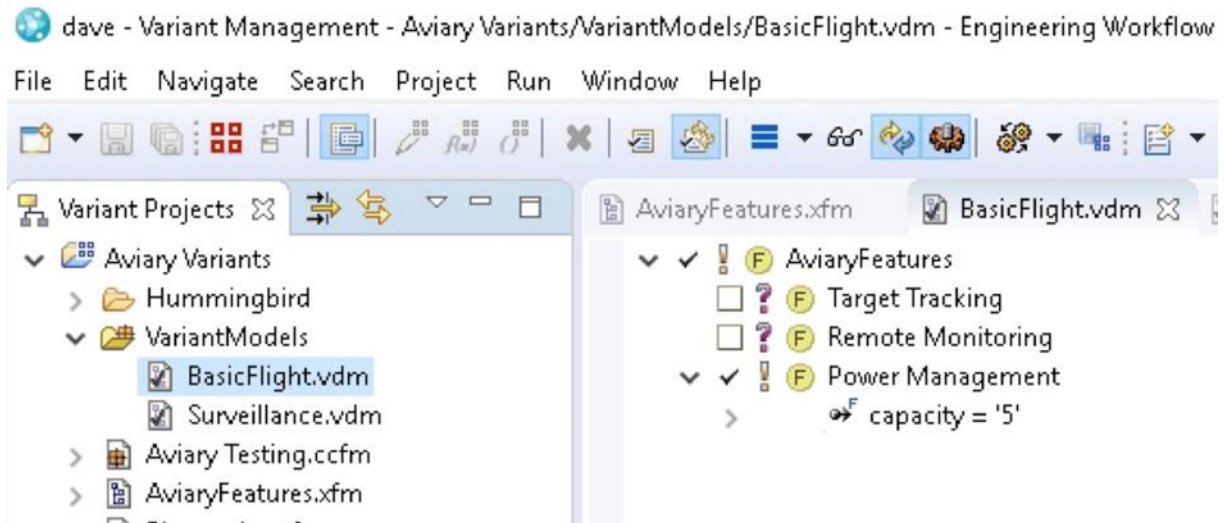


Beim Klick auf ein Feature in IBM EWM werden weitere damit verbundene Informationen angezeigt, die auch in anderen mit IBM EWM und pure::variants integrierten Tools liegen können. Das Feature „Target Tracking“ wird als optionales Feature angezeigt, das verlinkt ist mit dem verpflichtenden Feature „Power Management“ sowie mit der entsprechenden Anforderung in IBM Engineering Requirements Management DOORS Next und Testfällen in IBM Engineering Test Management:

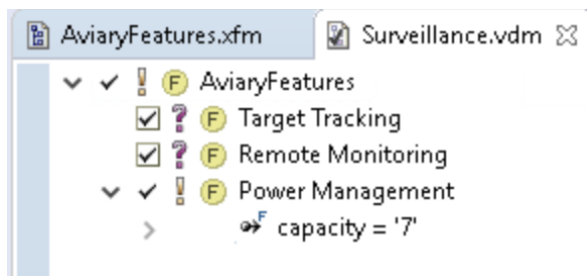


In diesem Beispiel sind zwei Varianten definiert. Beim Öffnen der Variante „BasicFlight“ wird angezeigt, welche Features zu dieser Variante gehören und welcher Wert variantenabhängigen Parametern jeweils zugewiesen wird. In diesem Fall sehen wir, dass das Feature „Power

Management“ zur Variante „BasicFlight“ gehört, und dass der Parameter „capacity“ für diese Variante auf den Wert 5 gesetzt wird:

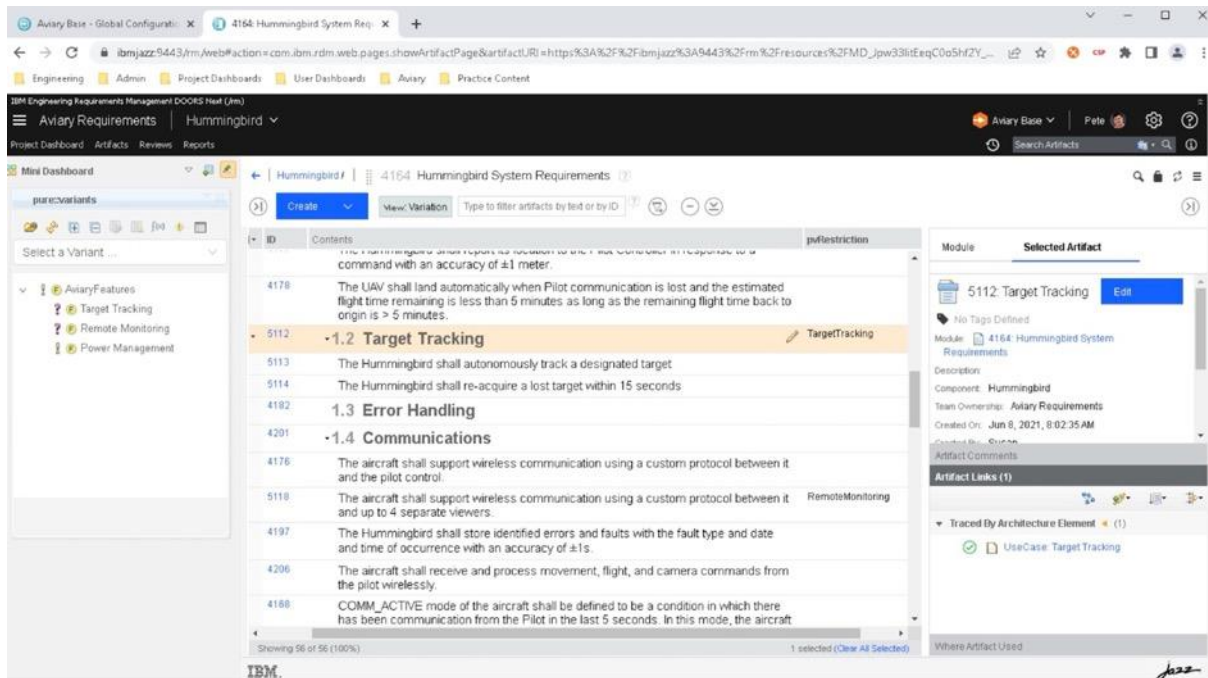


Bei Auswahl der Variante „Surveillance“ wird das optionale Feature „Target Tracking“ angezogen und der Parameter „capacity“ wird auf 7 gesetzt, da dies so in den Rahmenbedingungen (Constraints) des Feature-Modells hinterlegt ist.

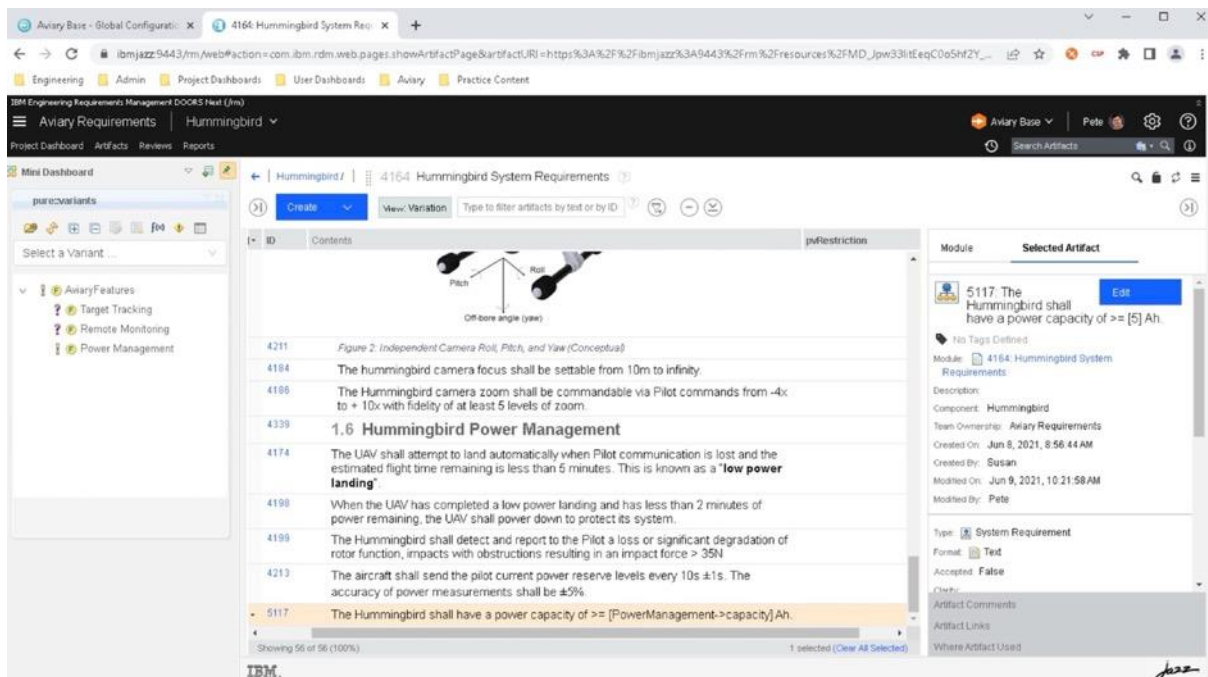


Die Varianten aus der IBM DOORS Next Perspektive

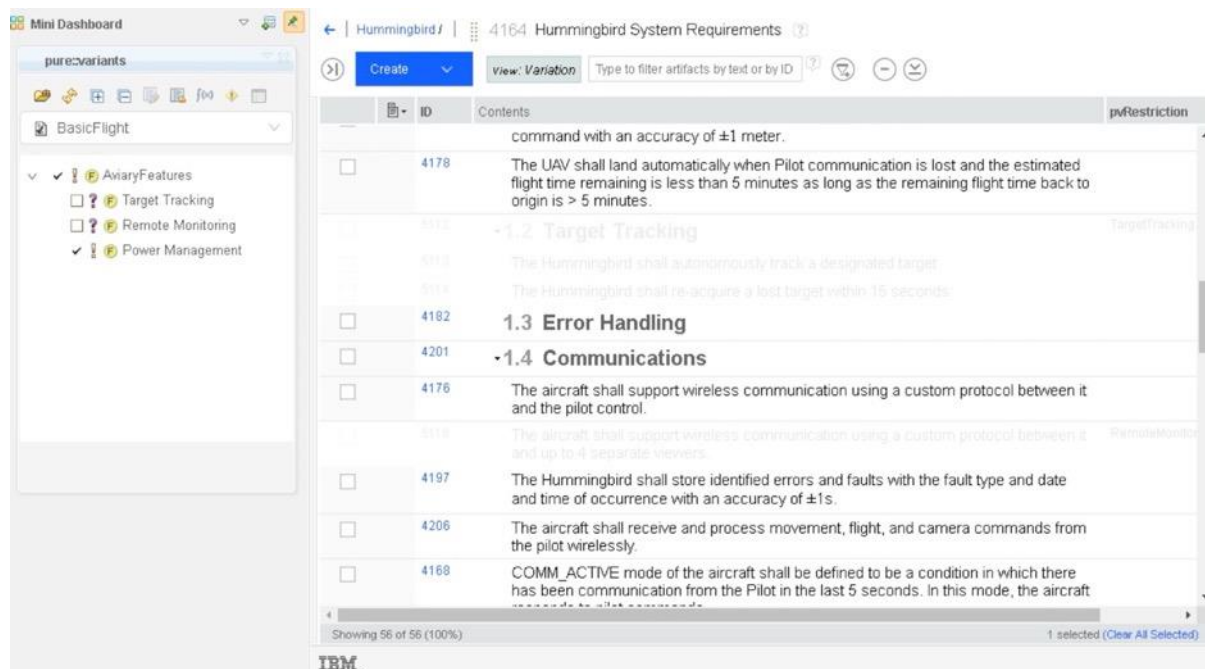
Aus der Perspektive eines IBM DOORS Next Anwenders sehen wir die Integration des pure::variants Feature-Modells als Widget eingebettet in das Mini-Dashboard auf der linken Seite des Fensters. Unter den Systemanforderungen finden sich die konkreten Anforderungen für das Feature „Target Tracking“. Bei Auswahl dieses Features im Anforderungsfenster wird rechts auf dem Bildschirm angezeigt, mit welchem Element im Architekturmodell dieses Feature beschrieben wird.



Eigenschaften, deren konkrete Auslegung von der gewählten Variante abhängt, können in IBM DOORS Next in Form von Platzhaltern hinterlegt werden, deren Wert sich dann mit Auswahl einer Variante definiert:

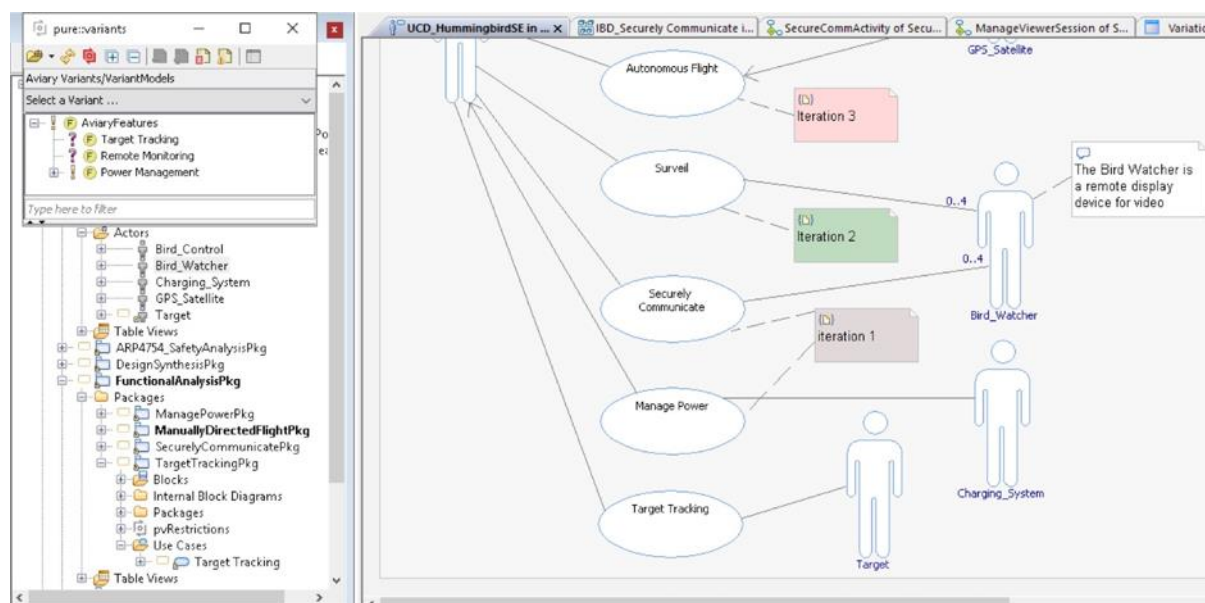


Bei Auswahl der Variante „BasicFlight“ ist das Feature „Target Tracking“ deselektiert, und daraus ergibt sich eine erforderliche Kapazität von 5Ah. Zudem sind die Anforderungen, die zum Feature „Target Tracking“ gehören, in der Ansicht der Anforderungen ausgegraut:

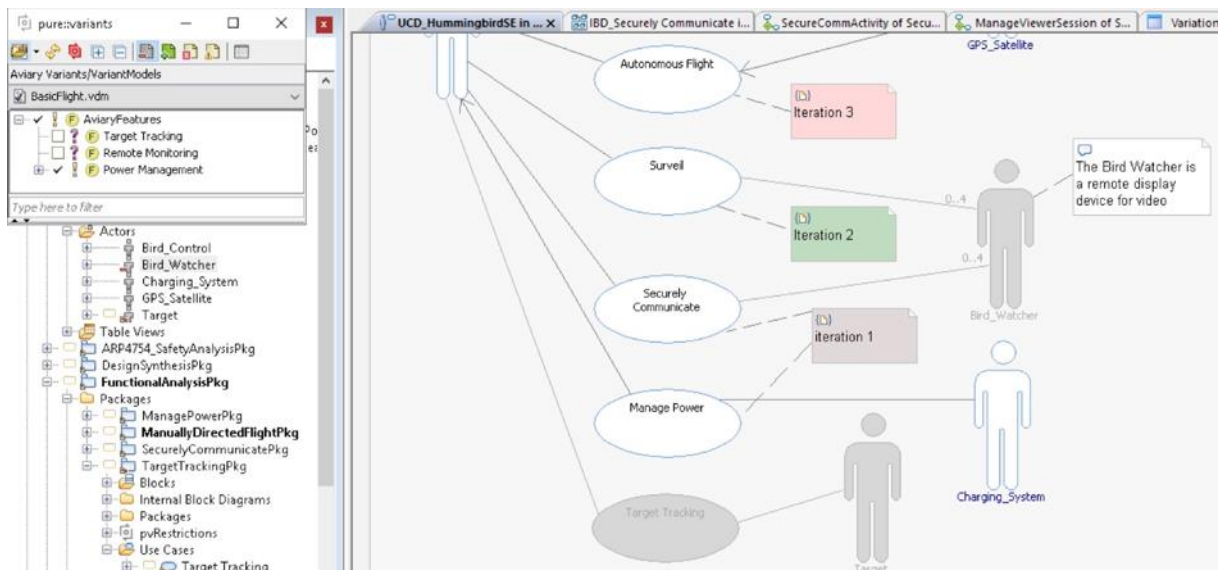


Handling der Varianten in IBM Rhapsody

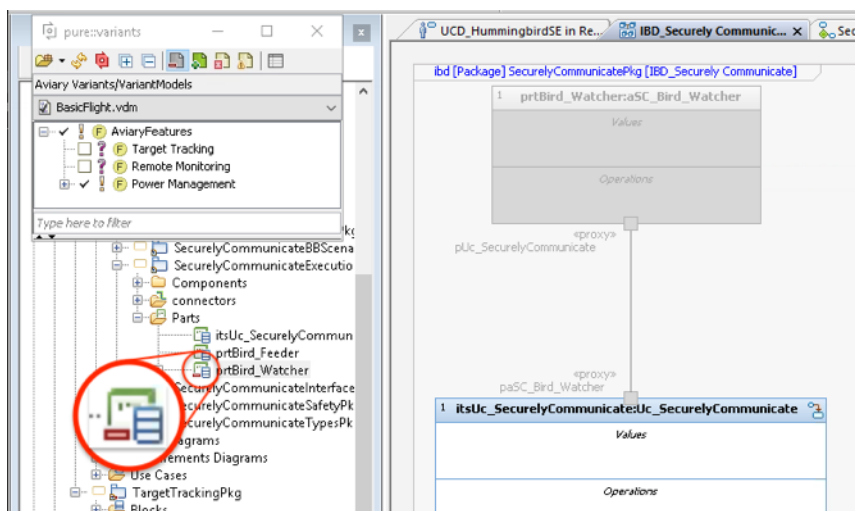
In IBM Rhapsody wird zunächst das komplette 150%-Modell dargestellt, inklusive aller optionalen Features. Über das Plug-in (links oben im Screenshot) sind die einzelnen Features sichtbar, und es kann eine Variante ausgewählt werden:



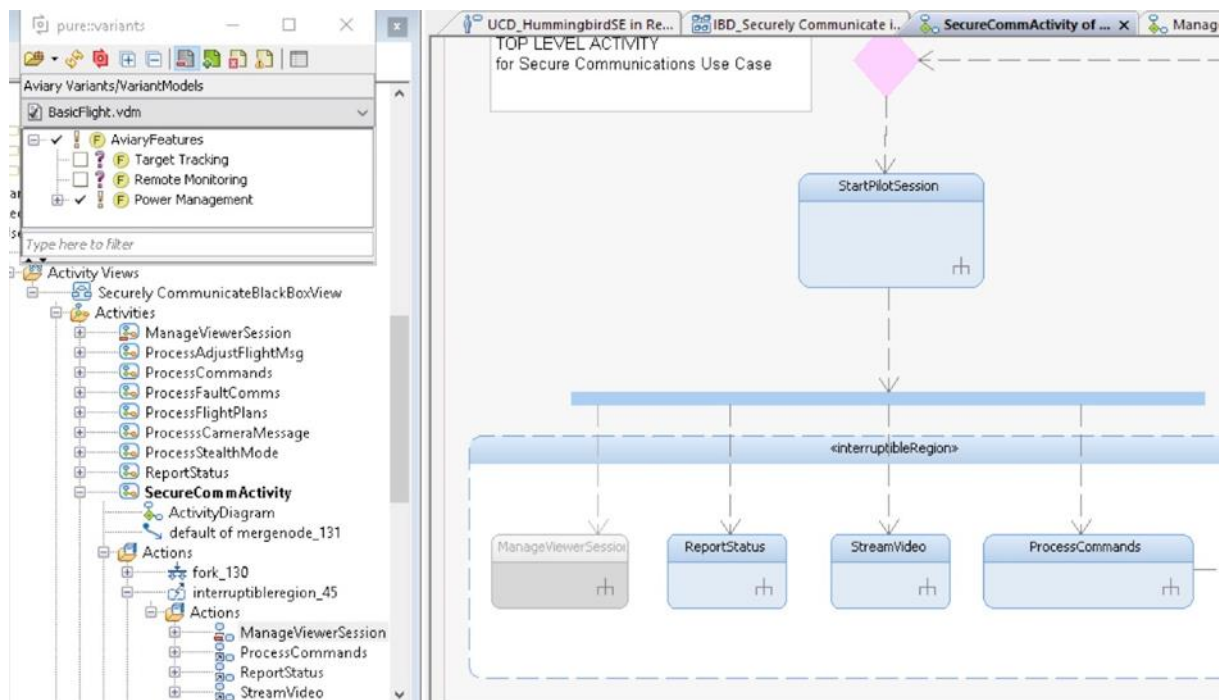
Wird nun die Variante „Basic Flight“ selektiert, werden damit automatisch die optionalen Features „Target Tracking“ und „Remote Monitoring“ deselektiert. Modellelemente, die zu den deselektierten Features gehören, werden im Modell ausgegraut:



Die Auswahl einer Variante und die sich daraus ergebenden Auswirkungen werden automatisch in allen Diagrammen reflektiert. Modellelemente des 150%-Modells, die in dieser gewählten Variante ausgeschlossen sind, werden auch im Modellbaum durch ein Minus im Icon gekennzeichnet, wie hier am Beispiel des Parts „prtBird_Watcher“:

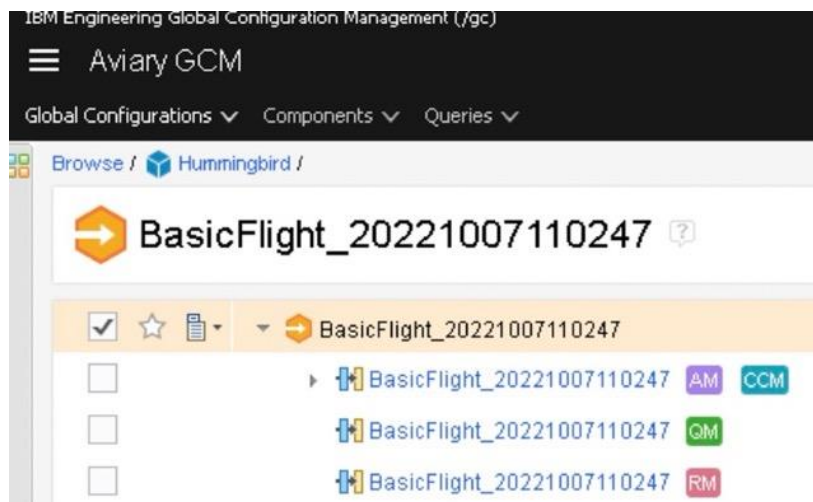


Ein weiteres Beispiel sehen wir hier bei der Aktion „ManageViewerSession“, die zum Feature „RemoteMonitoring“ gehört, welches in der Variante „BasicFlight“ ebenfalls nicht vorhanden ist:



Ausleitung der Variante „BasicFlight“

Aus IBM Engineering Global Configuration Management lässt sich nun automatisiert eine gewünschte Variante des Hummingbird ausleiten.

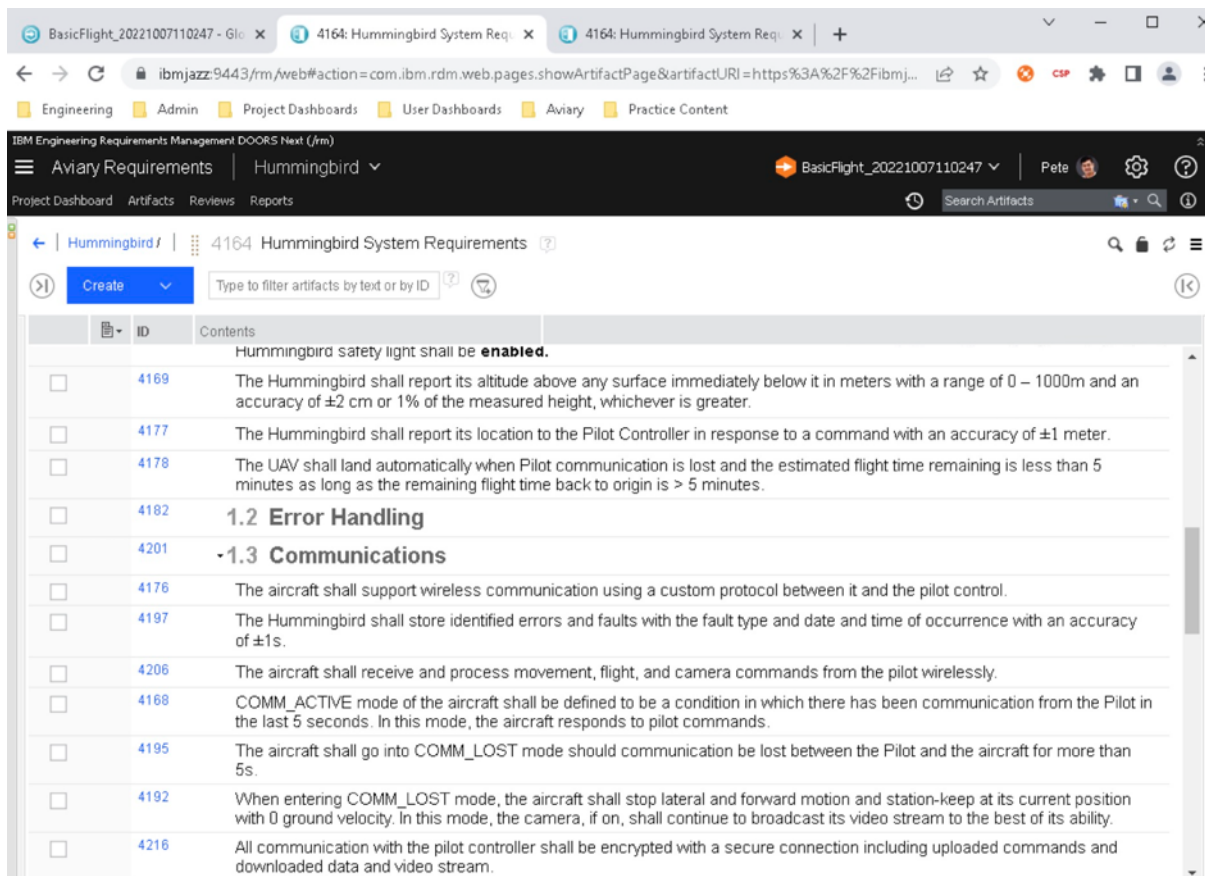


In unserem Beispiel werden mit der Ausleitung für die Variante „BasicFlight“ in einem übergreifenden Global Configurations Scenario die Streams erstellt für die involvierten Disziplinen. Die in diesem Beispiel ausgewählten Disziplinen sind:

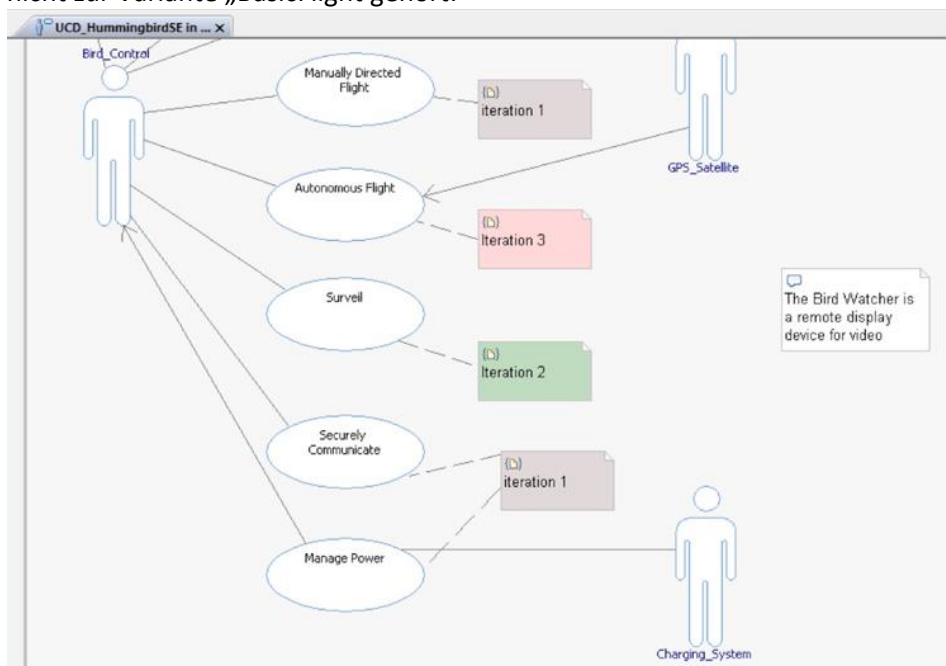
- CCM/AM (ChangeControlManagement, ArchitectureManagement), z.B. Ticketverwaltung, Architekturmodelle (z.B. in IBM Rhapsody), weitere Dateien und Dokumente nach Bedarf
- QM (Qualitätsmanagement), z.B. Testdurchführung, -abdeckung und -ergebnisse
- RM (Requirements Management), z.B. Anforderungsdokumente, Lastenhefte, IBM DOORS Next Module)

Dabei werden alle Elemente entfernt, die Features betreffen, welche in der Variante „BasicFlight“ nicht enthalten sind. So entsteht aus dem 150% Modell das 100% Modell, das exakt die selektierte Variante abbildet.

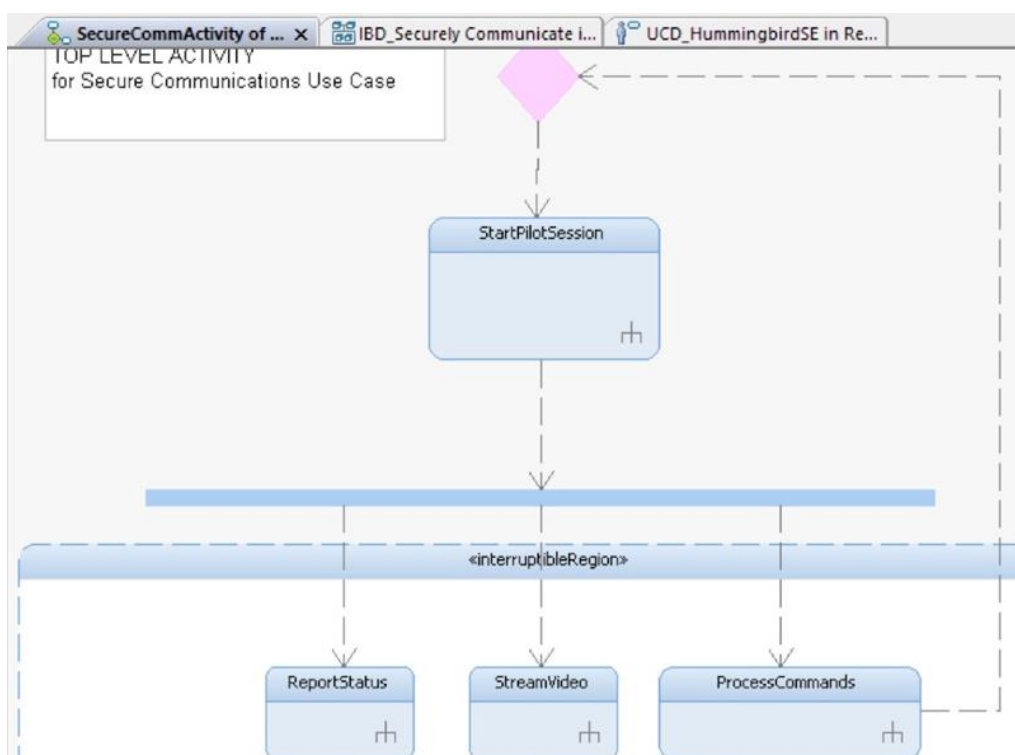
In IBM DOORS Next werden mit der Variante „BasicFlight“ die oben ausgegrauten Anforderungen nicht mehr angezeigt. Der Abschnitt „Target Tracking“ wurde entfernt:



Im UML Modell sind die mit „Target Tracking“ verbundenen Elemente entfernt, da dieses Feature nicht zur Variante „BasicFlight“ gehört:



Der UseCase Target Tracking und entsprechende Aktoren wurden entfernt, ebenso wie der Part prtBird_Watcher und die Aktion ManageViewerSession:



Einer der wichtigsten Vorteile dieser Herangehensweise, bei der alle Varianten in einem zentralen Tool verwaltet werden und darin zugänglich sind für alle Tools im Lifecycle, liegt in der Minimierung von Risiken:

- Jede Variante ist eindeutig definiert, Divergenz durch unterschiedliche Speicherorte und lückenhafte Sichtbarkeit ist ausgeschlossen.
- Zugriff auf das Variantenmanagement erfolgt aus den Expertentools, ein Wechsel des Tools im Arbeitsablauf ist nicht erforderlich, und damit auch keine spezifischen Toolschulungen für die Experten. Fehler aufgrund mangelnder Kenntnis oder Akzeptanz eines „fremden“ Tools sind ausgeschlossen.
- Die Varianten werden im zentralen Tool ausschließlich nach Feature-Aspekten definiert (inklusive Rahmenbedingungen und Einschränkungen), eine ungewollte Vermischung von Varianten, Versionen, Releases, etc. wird vermieden.

Zudem sind die abgeleiteten Varianten optimiert bezüglich der enthaltenen Elemente. Es wird kein „Ballast“ mitgeführt, der unnötig Speicher- oder Rechenkapazität braucht. Diese optimierten Modelle eignen sich auch sehr gut zur Weitergabe an Projektpartner oder Zulieferer, die damit ausschließlich die Inhalte der relevanten Variante erhalten.

Im Vergleich zu den in früheren Ausgaben bereits betrachteten Alternativen „Spezielles UML Profil“ und „UML VariationPoint“ empfiehlt sich die hier beschriebene Vorgehensweise insbesondere für Anwendungsbereiche, in denen zur Übergabe an Projektpartner jeweils eine konsistente Ausleitung verschiedener Disziplinen benötigt wird, die frei von Inhalten aus anderen Varianten ist. Das ist regelmäßig der Fall, wenn in einem Hersteller-/Zulieferer-Szenario verschiedene Produktlinien mit Feature-Sets zu managen sind, die teilweise überlappen oder disjunkt sind. Das Einbeziehen neuer Features lässt sich ebenfalls leicht realisieren, genauso wie das sukzessive und kontrollierte „Herausnehmen“ von Features, die nicht mehr erforderlich sind.

Fazit:

Wir haben in dieser Serie drei unterschiedliche Varianten für das Management von Varianten bei Modell-basierter Entwicklung vorgestellt. Welche Option im konkreten Projekt zum Einsatz kommt, hängt von den Rahmenbedingungen ab; es gibt keine „beste Lösung“, die für jedes Projekt gleichermaßen optimal funktioniert.

1. Nutzung der Funktionen, die im Modellierungswerkzeug vorhanden sind: In dem UML-Modellierungswerkzeug IBM Rhapsody gibt es die Möglichkeit, Varianten über das Element VariationPoint im Modell zu definieren. Diese Option empfiehlt sich für Projekte mit Implementierungsvarianten, die keine oder nur überschaubare Abhängigkeiten zueinander haben und durch das Austauschen einzelner Klassen oder Blöcke realisiert werden.
2. Nutzung eines speziellen UML-Profiles: Mit dem VAMOS Profil von Tim Weillkiens, das spezifisch für Variantenmanagement im Systems Engineering ausgelegt ist, können Variantenmodelle präzise modelliert werden, inklusive aller Elemente, die zu anderen Varianten gehören (150%-Modell) sowie definierte Konfigurationen dieser Varianten innerhalb des SysML-Modells. Diese Option ist empfehlenswert für komplexe Varianten, die dennoch vollständig im SysML/UML-Modell beschrieben werden sollen.
3. Nutzung eines spezialisierten Tools für Variantenmanagement: Ein Tool wie pure::variants lässt sich nahtlos in die Entwicklungsumgebung integrieren und ermöglicht die zentrale

Definition und Verwaltung von Varianten und damit verbundenen Features. Diese Option ist von Vorteil, wenn die Bildung von Varianten sich durch alle Disziplinen im Entwicklungsprozess zieht und daher alle Beteiligten Zugriff auf Varianten und Features haben müssen. Die Möglichkeit, ausschließlich optimierte 100%-Modelle für die jeweiligen Disziplinen auszuleiten, um sie an Projektpartner zu übergeben, sorgt zudem für eine Verschlankung und Vereinfachung der Zusammenarbeit im Projekt.

Im Rahmen dieser Artikelserie war natürlich nur eine relativ oberflächliche Betrachtung der Optionen möglich; eine detailliertere Beschreibung hätte den Rahmen gesprengt. Falls Sie sich gerade mit dem Thema Variantenmanagement beschäftigen und tiefer in die Materie eintauchen wollen, sprechen Sie uns jederzeit an. Das hier beschriebene Beispiel können Sie selbst anschauen und sich damit in einer Sandbox-Umgebung vertraut machen; die Zugangsdaten schicken wir Ihnen gern auf Anfrage. Auch für einen unverbindlichen ersten Austausch stehen unsere Experten gern zur Verfügung. Nehmen Sie Kontakt auf per Email an info@sodiuswillert.de, und wir kommen ins Gespräch.