

# Blunter 4.0 - C# to C++ Transpilation

## Über yWorks GmbH

Die yWorks GmbH ist weltweit führender Anbieter für Softwarebibliotheken zur Visualisierung und Bearbeitung von Graphen, Diagrammen und Netzwerken. Die größten Softwareunternehmen weltweit setzen unsere Bibliotheken ein, um ihrerseits Programme zu erstellen, mit denen strukturierte Daten automatisch erstellt, bearbeitet, angeordnet und dargestellt werden. Unsere kommerziellen Lösungen bieten die umfassendsten Algorithmen zur automatischen Anordnung von strukturierten Daten in Diagrammform. Die Softwarebibliotheken sind für verschiedene Programmierplattformen und -sprachen erhältlich und werden aufgrund ihrer Vielseitigkeit in sämtlichen Industrie- und Wirtschaftsbereichen genauso eingesetzt wie in der Forschung.

## Hintergrund

Unser Kernprodukt „yFiles“ ist eine Programmierbibliothek, die Softwareentwickler für ihre eigenen Applikationen einsetzen, um komplexe Diagramme und Graphen einfach darzustellen. Ein Schwerpunkt des Produkts sind die automatischen Layoutalgorithmen, die dafür sorgen, dass große strukturierte Daten automatisch in lesbare Diagramme umgewandelt werden können. Algorithmen aus dem Bereich Graphenzeichnen werden hierfür von yWorks originär lediglich in einer einzigen Programmiersprache implementiert. Um diese Algorithmen auch auf anderen Plattformen direkt verfügbar zu machen, besitzt yWorks Hilfsmittel, um den Quelltext dafür automatisch in Quelltext anderer Programmiersprachen zu überführen. Eine besondere Herausforderung bei diesem Übersetzungsvorgang ist, dass das entstehende Resultat nicht nur lauffähig ist, sondern handgeschriebenem Code möglichst nahekommt. Das bedeutet unter Anderem, dass es möglich sein soll, den Code nach der Übersetzung weiter manuell zu bearbeiten und direkt zu debuggen. Viel wichtiger ist aber, dass Konventionen, Normen, Konzepte, und möglichst auch Basisbibliotheken weitestgehend auf der Zielseite eingehalten bzw. wiederverwendet werden, damit Programmierer, die mit dem Resultat auf der Zielplattform arbeiten, das Gefühl mit einer „nativen“ Bibliothek zu arbeiten vermittelt wird.

Ein solches Übersetzungstool (interner Codename „Blunter“) setzen wir ein, um Code der in C# geschrieben wird sowohl nach Java, als auch alternativ nach JavaScript zu übersetzen. Dieses Werkzeug ist vollständig bei uns mithilfe der Sprache C# entwickelt worden. Das Tool nutzt die Open-Source .NET Compiler Plattform „Roslyn“ um C# Code zu parsen und zu analysieren. Es ist sowohl unter Windows als auch unter Linux und Mac lauffähig.

Im Jahr 2018 konnten wir bereits mit Studenten im Rahmen des Tübinger Software Projekts 2018/2019 die Sprache Python als drittes Ausgabeformat erfolgreich hinzufügen.

Im Zuge des Tübinger Software Projekts in diesem Jahr soll das Übersetzungstool „Blunter“ diesmal derart erweitert werden, dass es als weitere Sprache C++ 11 bei der Ausgabe unterstützt. Das finale Ziel hierbei ist es eine Version unserer Klassenbibliothek in C++ zu erzeugen, die komfortabel von Entwicklern dieser Sprache genutzt werden kann. Erste Schritte auf dem Weg zu diesem Ziel haben wir in einem internen Prototyp bereits in einem Testprojekt erzielt. Bis unsere komplette Bibliothek aber lauffähig übersetzt werden kann, fehlen noch einige wichtige Aspekte. Dazu muss z.B. festgelegt

werden, wie besondere Sprachunterschiede im Bereich Generics/Templating aufeinander abgebildet werden sollen oder können und wie das Problem der „Garbage Collection“ in C++ möglichst automatisch und effizient gelöst werden könnte. Für yWorks speziell ist es hier nicht notwendig, dass sämtliche Eigenschaften der Ursprungssprache C# abgedeckt werden. Für unseren Anwendungsfall haben wir uns im Ausgangsprodukt auf eine Untermenge der vielfältigen Möglichkeiten von C# beschränkt und weniger häufig genutzte, sehr spezielle Fähigkeiten der Sprache bewusst ausgelassen, um die Übersetzung zu anderen Sprachen zu vereinfachen.

Nach der Recherche zu den Unterschieden der Sprachen C# und C++ 11 werden in den folgenden Schritten die Konvertierungslogik implementiert und mit Tests die Ergebnisse überprüft. Für die Konvertierung bietet sich hier eine schrittweise Herangehensweise mit mehreren Milestones an, bei der z.B. initial noch nicht auf identisches Laufzeitverhalten und Speicherverbrauch geachtet wird oder Teile zwar lauffähig aber nicht den Plattformnormen entsprechend implementiert werden. Nach und nach können dann die verschiedenen offenen Fragestellungen angegangen und einzeln gelöst werden.

Für die Entwicklung des Übersetzers kann bevorzugt JetBrains Rider auf allen der drei großen Betriebssystemplattformen oder Visual Studio unter Windows eingesetzt werden. Während der Entwicklung sollte continuous testing und continuous integration eingesetzt und eine hohe Testabdeckung durch Integrationstest erreicht werden.

Das Projekt wird mit den agilen Methoden der Softwareentwicklung durchgeführt (SCRUM). Dies umfasst alle Schritte von der Festlegung der Roadmap, über die Sprintplanung bis zur Durchführung der Implementierung und der Tests.

Wir erhoffen uns aufgrund des anspruchsvollen aber sowohl theoretisch als auch aus praktischer Sicht spannenden Themas eine aktive Mitarbeit. Die Möglichkeit, gestaltend und mit viel Selbstbestimmung an modernster Technologie mitzuwirken verspricht gleichzeitig Spaß und Herausforderungen.

Der Kick-Off wird in den Räumen der yWorks GmbH in Tübingen stattfinden. Mitarbeiter der yWorks GmbH, welche das bestehende Projekt implementiert haben und bereits die Gruppe des Tübinger Software Projekts aus dem letzten Jahr erfolgreich begleiteten, werden euch vor Ort die Geschichte hinter den Projekten und das Produkt selbst vorstellen. Sie stehen nachfolgend wie auch im letzten Jahr während des ganzen Projekts beratend als Ansprechpartner zur Verfügung.

## Überblick

### Zielsetzung

- Weiterentwicklung des Prototyps der C++ Version des bestehenden C# Sourcecodetranspilers „Blunter“
- Bestimmung der syntaktischen und semantischen Unterschiede zwischen den Sprachen C# und C++ 11 und den Grundbausteinen der jeweiligen Kernbibliotheken und Funktionalitäten.
- Festlegung einer Abbildung der unterschiedlichen Funktionalitäten der Sprache C# auf die Sprache C++
- Anpassung und Erweiterung des bereits bestehenden Ausgabesyntaxbaumes für C++
- Erweiterung der Übersetzerfunktionalität
- Überprüfung der Übersetzung mittels Integrationstests

## Anforderungen

- Gute Erfahrungen in objektorientierter Programmierung
- Erfahrungen mit C# und/oder C++ 11
- Interesse an Programmiersprachen und gutes analytisches Denken sind von Vorteil.  
Erfahrung in Graphentheorie oder mit speziellen Algorithmen ist nicht notwendig.
- Implementierungssprache und vorhandener Code: C#
- Sourcecodeversionierung: Git
- Abdeckung der Anforderungen durch Unit Tests
- Entwicklungssysteme: Windows, Linux, Mac je nach Vorliebe
- Entwicklungsumgebungen: bevorzugt JetBrains Rider oder Visual Studio mit Erweiterungen
- Continuous Integration: Jenkins oder Atlassian Bamboo
- Teamkommunikation: Slack oder Ähnliches je nach Vorliebe
- Sprintplanung und Durchführung mit Atlassian JIRA
- Begleitende Dokumentation mit Atlassian Confluence