**GTlab - IntelliGraph**

**Einleitung**

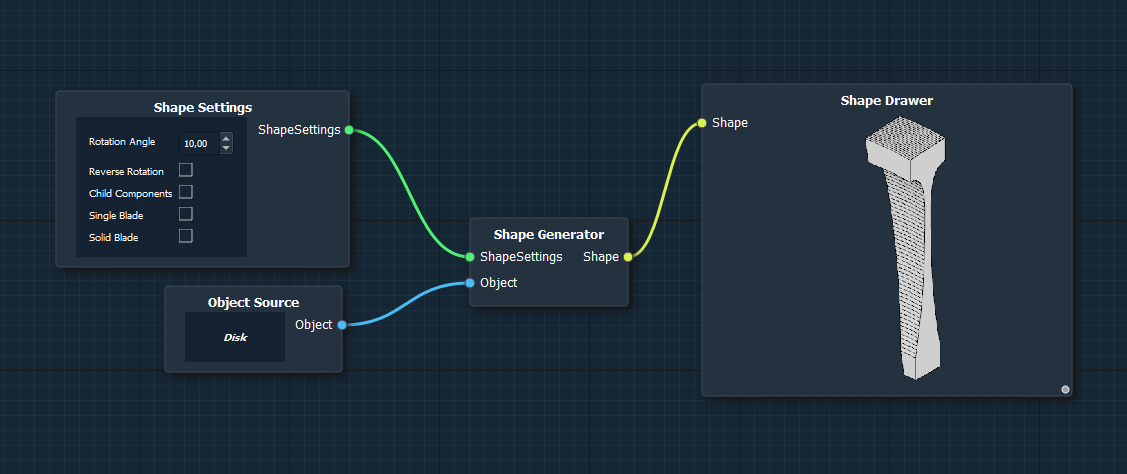
Effiziente Verarbeitung, Analyse und Visualisierung von Daten in einer flexiblen, modularen und benutzerfreundlichen Umgebung. Das System ist darauf ausgelegt, sowohl für Anfänger als auch für fortgeschrittene Benutzer leicht zugänglich und anpassbar zu sein. Die node-basierte Architektur ermöglicht, komplexe Workflows und Analyseprozesse auf einfache und intuitive Weise zu erstellen und zu verwalten.

Die Hauptfunktionalität eines IntelliGraph besteht darin, Daten von Node zu Node zu leiten, wobei jeder Node eine bestimmte Operation oder Verarbeitung auf den Daten ausführt. Nodes können sowohl zur Visualisierung von Daten, wie z. B. das Anzeigen von Triebwerksdaten oder anderen komplexen Objekten, als auch zur Durchführung von Berechnungen, Filterungen und Transformationen verwendet werden.

Ein IntelliGraph kann unter bestimmten Voraussetzungen selbst als eine Node (IntelliGraph-Node) in einem anderen IntelliGraph genutzt werden. Dies ermöglicht die Erstellung von hoch-komplexen Workflows ohne den Verlust der Übersichtlichkeit. Darüber hinaus können über so genannte DataProvider IntelliGraph-spezifische Eingabemasken und Symbolbilder automatisiert erzeugt und in die IntelliGraph-Node integriert werden.

Mit der Fähigkeit, benutzerdefinierte Nodes über GTlab-Module bereitzustellen, kann sich der IntelliGraph spezifischen Anforderungen anpassen. Darüber hinaus lassen sich IntelliGraphs einfach serialisieren und sind deshalb sehr portabel. Innerhalb des GTlab Frameworks können projekt-spezifische IntelliGraphs mit anderen geteilt werden.

Jedem IntelliGraph kann eine Beschreibung basierend auf Markdown angehängt werden. Ebenfalls besteht die Möglichkeit MetaDaten zu hinterlegen, um z.B. Informationen zu In- und Outputs oder im Kontext der IntelliGraph-Node (Größe, Symbolbild, etc.) zur Verfügung zu stellen.



**Datenstruktur**

<https://github.com/hosseinmoein/DataFrame/wiki>

Generell können die Nodes innerhalb eines IntelliGraph verschiedenste Arten von Daten austauschen. Für allgemeine Filterung und Transformation eignet sich das Object-System von GTlab, erfordert jedoch eine Modifikation bezüglich der Handhabung innerhalb und zwischen den Nodes. Die interne Datenstruktur eines IntelliGraph sollte in der Lage sein, den gesamten zugehörigen Datensatz (DataSet) zu speichern und bei Bedarf an den nächsten Knoten weiterzugeben.

Ein DataItem ist die grundlegende Einheit und repräsentiert einen individuellen Datensatz. Ein DataItem kann unterschiedliche Datentypen enthalten, wie z. B. numerische Werte, Text, benutzerdefinierte Objekte, GtObjects oder komplexe Datenstrukturen wie Geometrien.



Ein DataSet ist eine Sammlung von DataItem-Objekten und stellt eine Gruppe verwandter Daten dar, die gemeinsam verarbeitet oder analysiert werden sollen.



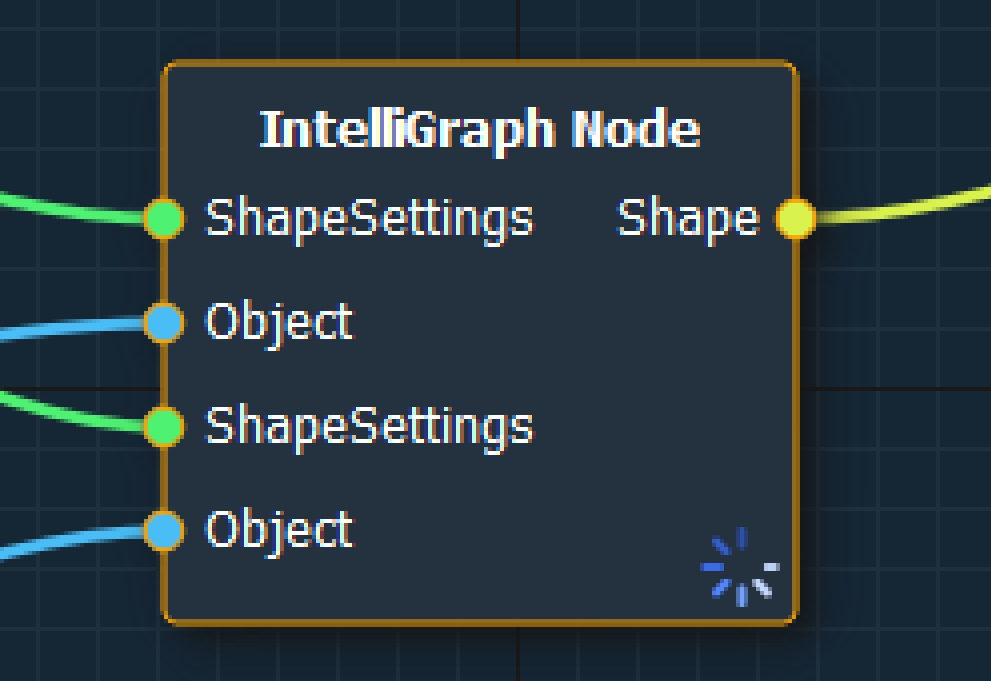
**Serialisierung**

Die Serialisierung eines IntelliGraph basiert auf dem Memento-System von GTlab.



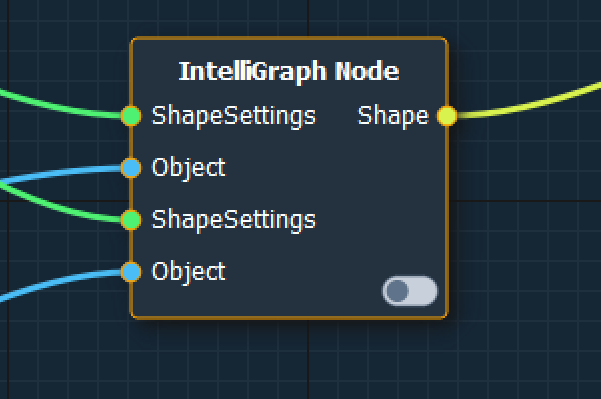
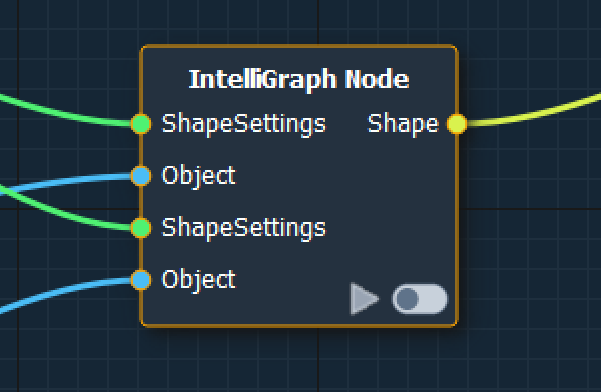
**Non-Blocking Architektur**

IntelliGraph ermöglicht es, Berechnungen innerhalb von Nodes asynchron durchzuführen, ohne die Hauptanwendung zu blockieren. Dies ist speziell bei aufwendigen Berechnungen wichtig, damit die Benutzeroberfläche flüssig und ansprechbar bleibt.

****

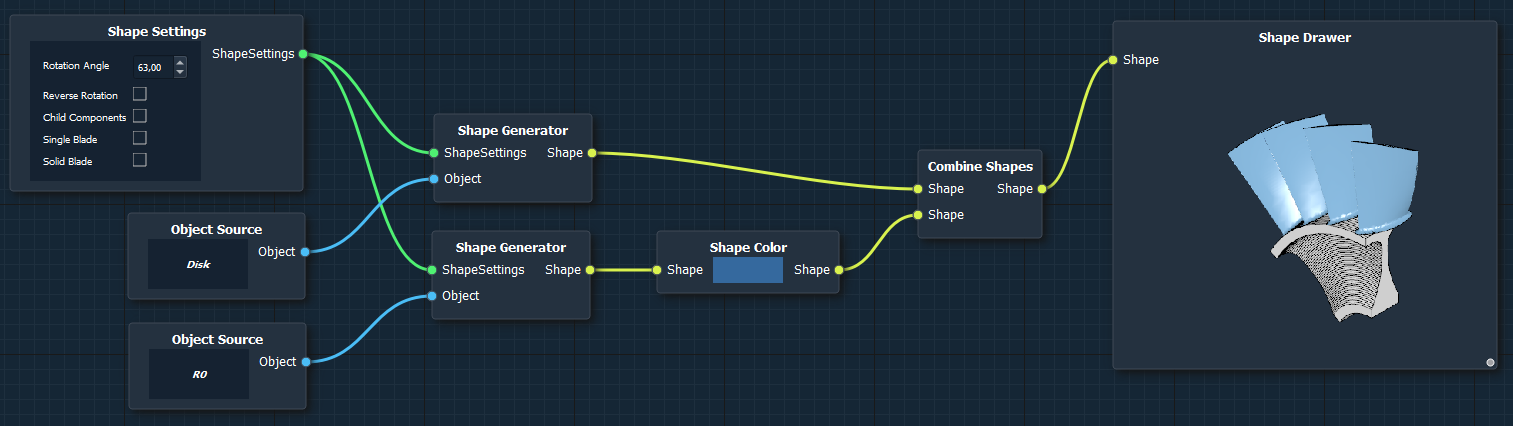
**Manuelle Ausführung**

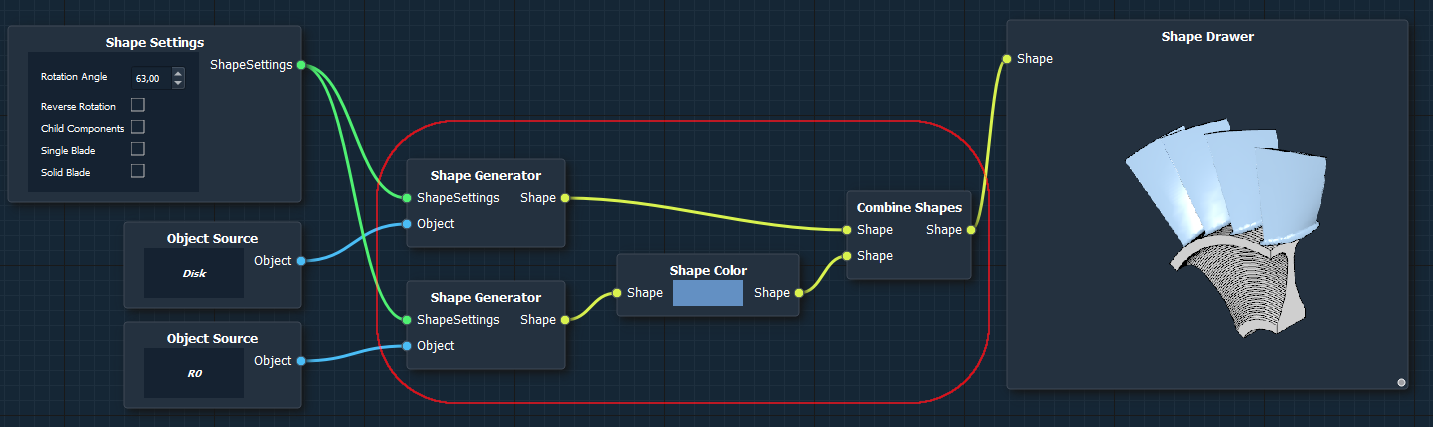
Standardmäßig werden Berechnungen innerhalb der Nodes im IntelliGraph automatisiert durchgeführt. Manchmal ist es sinnvoll die Ausführung von Nodes nur manuell durchzuführen. Innerhalb einer Node kann ein Slider genutzt werden, um die automatische Ausführung zu unterbinden. Wenn eine Node im IntelliGraph getriggert wird, erscheint innerhalb der Node eine Schaltfläche zur manuellen Ausführung der Node. Erst nach Betätigung der Schaltfläche und der Ausführung der Node, werden auch die Nodes stromab getriggert.

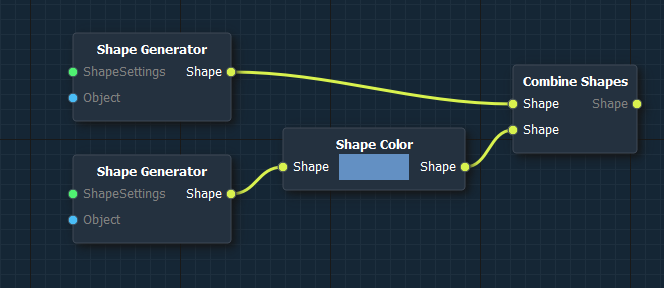
**** ****

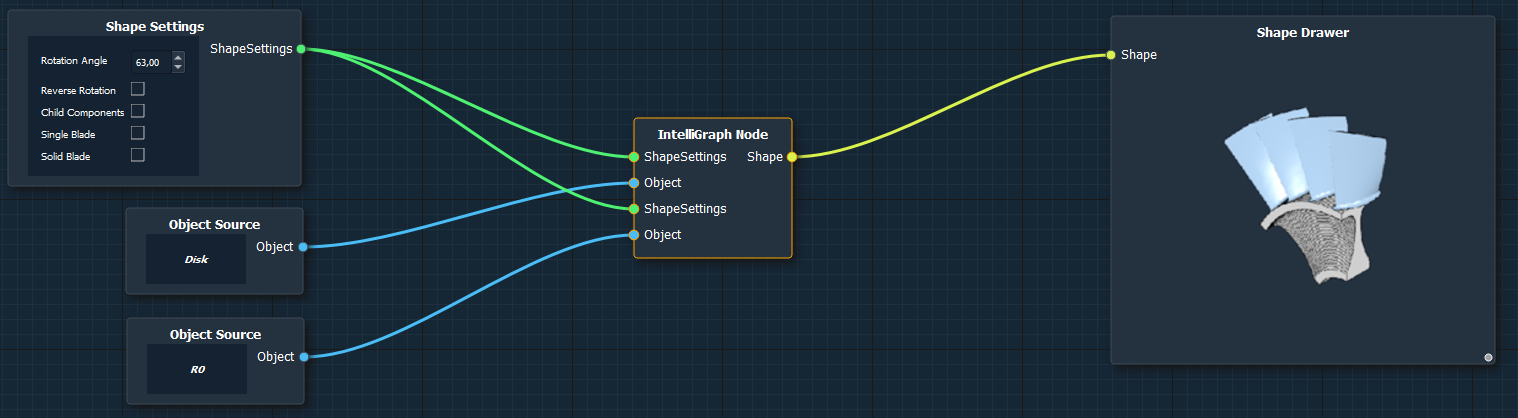
**IntelliGraph-Node**

Ein IntelliGraph kann als eigenständiger Node in einem anderen IntelliGraph genutzt werden.









**Task-Node**

xxx

**Python-Node**

xxx

**Beispiel-Prozess**

Standardmäßig werden Berechnungen innerhalb der Nodes im IntelliGraph automatisiert durchgeführt. Manchmal ist es sinnvoll die Ausführung von

* Auswahl einer GtpPerformanceTable mit verschiedenen GtpOperationgPointData Objekten die repräsentativ für verschiedene Triebwerksdesigns stehen (ObjectSource) und Aufbau eines entsprechenden DataSets.
* Plotten der verschiedenen Triebwerksdesign basierend auf auswählbaren Parametern in einer BarChartNode. Z.B. Schub.
* Möglichkeit zur Definition von Min und Max für den entsprechenden Parameter durch den Benutzer oder benutzerdefinierte Auswahl spezifischer Triebwerksdesign aus dem Plot
* Automatisierte Filterung basieren auf Usereingaben innerhalb der BarChartNode und Weitergabe der gefilterten Triebwerksdesigns an die nächste Node
* Durchführung eines Tasks zur Erstellung einer initialen Geometrie des Triebwerks innerhalb einer Task-Nodes und Hinterlegung einer Geometrischen Größe (z.B. Höhe der letzten Verdichterschaufel) im DataSet (Erweiterung des DataSets)
* Node zur Filterung spezifischer Designs bezüglich max. Schaufelhöhe