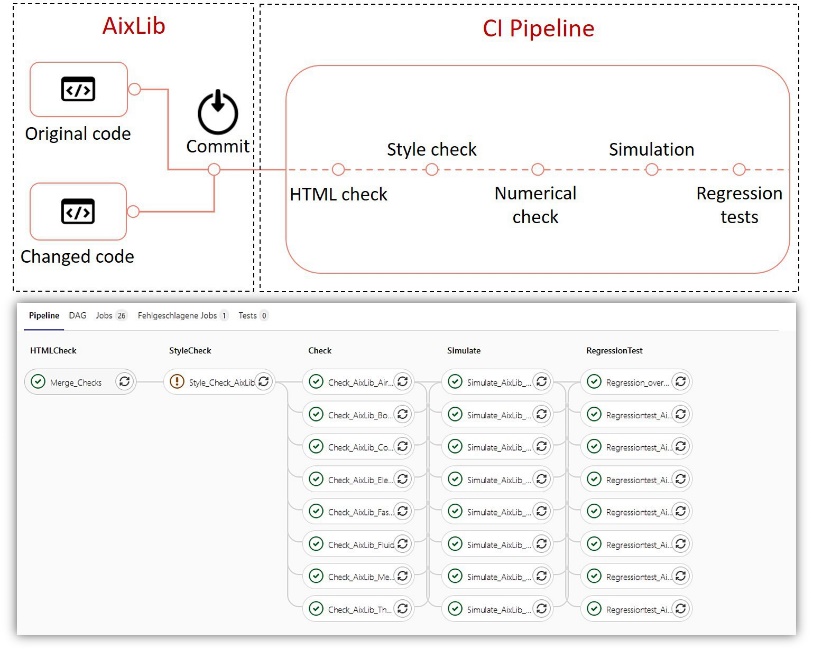
# Continuous Integration (CI):

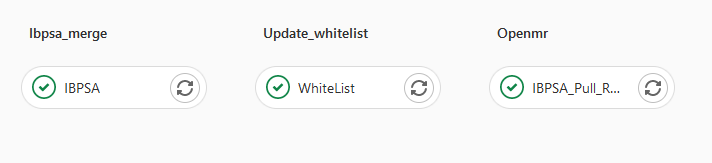
Bei der Modellierung komplexer Systeme arbeiten oft mehrere Beteiligte an einem Projekt. Code-Änderungen können verheerende Auswirkungen auf die Ergebnisse der aggregierten Modelle haben. Mit Hilfe von Continuous Integration (CI) wird die Softwarequalität unserer Open-Source-Bibliothek AixLib sichergestellt. Dazu werden die in der Sprache Modelica geschriebenen dynamischen Simulationsmodelle kontinuierlich auf Servern mit Testskripten und Simulationen auf Semantik und Ergebnistoleranz geprüft. Kürzlich haben wir das ursprüngliche Framework mit weiteren CI-Tests umfangreich erweitert.

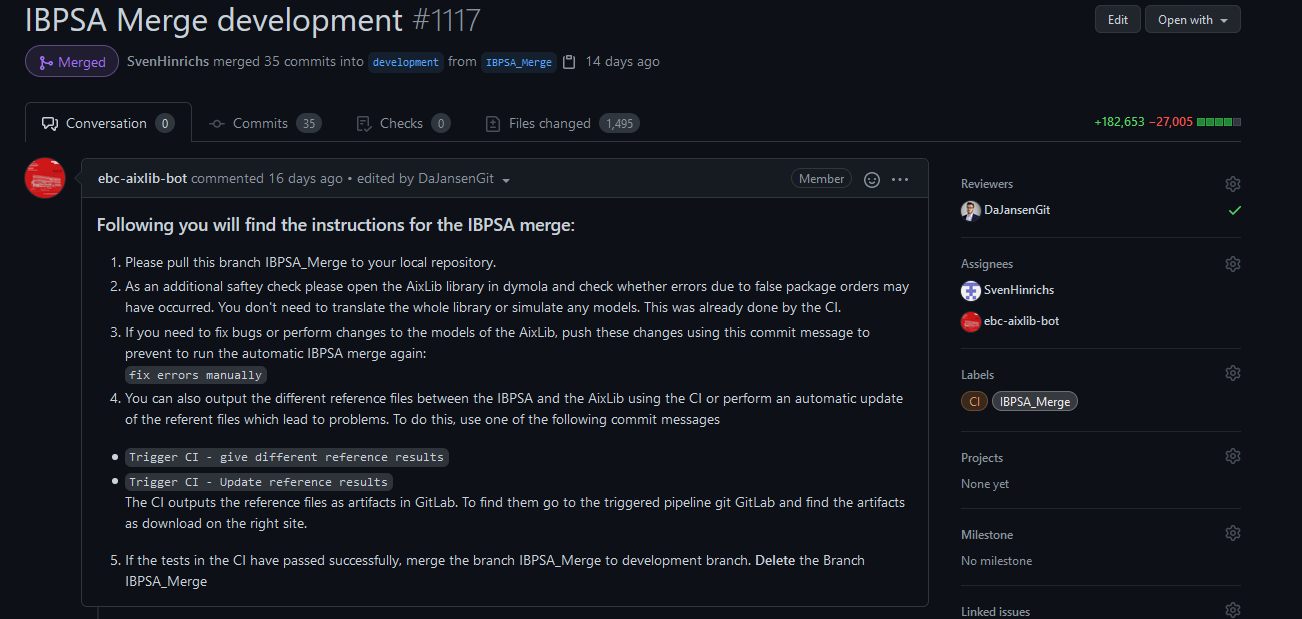
* **HTML-Check**
  + Um einen Modellierer auf Fehler in der HTML-Syntax hinzuweisen, wird ein HTML-Check durchgeführt, der den HTML-Code überprüft und eventuelle Fehler automatisch korrigiert.
* **Style-Check** 
  + Bei dieser Prüfung werden die Modelle auf ihre Code-Konventionen überprüft, um einen einheitlichen Programmierstil zu gewährleisten.
* **Numerical check of models**
  + Die Modelle werden auf Vollständigkeit, numerische Korrektheit und Durchführbarkeit geprüft. Sollte dies bei einem Modell nicht der Fall sein, wird der Anwender über die Fehlerstelle informiert, so dass Änderungen leicht vorgenommen werden können.
* -**Simulation of complex system models**
  + Komplexe Systeme ("Beispiele") werden simuliert, um Gültigkeit, Zuverlässigkeit und Funktionalität sicherzustellen.
* **Regression tests**
  + Mit der Integration von Regressionstests aus BuildingsPy werden ausgewählte Beispiele simuliert und mit gespeicherten Daten aus Referenzergebnissen verglichen. Damit wird sichergestellt, dass Änderungen an Modellen in bereits getesteten Systemen oder vererbenden Klassen keine neuen Fehler verursachen und nicht von der vorgegebenen Toleranz abweichen.

Die beschriebenen Tests werden automatisch nacheinander in einer Pipeline durchgeführt, sobald überarbeiteter Quellcode hochgeladen (gepusht) wird. Auf diese Weise weist das CI den Modellierer bereits während des Modellierungsprozesses auf mögliche Fehler hin. Auf diese Weise spart der Entwickler viel Zeit bei der Fehlersuche. Fehlerhafte Änderungen werden so serverseitig blockiert. Das reduziert die Entwicklungszeit und erleichtert die detaillierte Fehlersuche in Modellen. Die umfangreichen Erweiterungen des CI-Frameworks sorgen für die kontinuierliche Steigerung der Qualität der AixLib. Im Rahmen unserer anspruchsvollen Qualitätssicherung werden derzeit fünf Tests durchgeführt, die entweder auf modifizierte Modelle oder auf vollständige Modellpakete in der Bibliothek angewendet werden.

# Continuous Integration (CI): Tests und Hilfsmittel

**IBPSA Merge**

Mithilfe der CI kann der aufwendige Vorgang des „IBPSA-Merge“ erleichtert und teilweise automatisiert ablaufen. Damit das dafür zugrundeliegende Template aufgerufen wird, muss im AixLib Repo ein neuer Branch „*IBPSA\_Merge*“ erstellt werden. Der IBPSA Merge erfolgt in drei unterschiedlichen Stufen 

* ***IBPSA\_Merge*** : Führt die aktuelle IBPSA Version und die neuste development AixLib Version zusammen (mergeModelica.py) . Anschließend wird ein neues Conversion Skript auf Basis des IBPSA Repo für die AixLib erstellt und in die [package.mo](http://package.mo) der AixLib eingebunden.
* ***Update\_whitelist***: Aktualisiert die WhiteLists für die CI
* ***Openmr***: Stell einen Pull Request von “IBPSA\_Merge” in “development”.
* 
* Die AixLib wird nun nocheinmal vollständig getestet und geprüft. Wenn die Pipeline der CI erfolgreich durchgelaufen ist, kann die neuste IBPSA Version in den *development* Branch gemerged werden.

***Erstellung eines Pull Request***

* Wird ein Pull Request erstellt, werden die 5 bereits erwähnten Stufen und dessen Jobs durchgeführt. (*HTML-Check*, *Style-Check* ***Numerical check of models, Simulation of complex system models, Regression tests).* Die Tests werden auf alle Modelle der AixLib angewendet.**

***Arbeiten auf dem Branch***

* Wird ein Pull Request erstellt, werden die 5 bereits erwähnten Stufen und dessen Jobs durchgeführt. (*HTML-Check*, *Style-Check* ***Numerical check of models, Simulation of complex system models, Regression tests).* Die Tests nur auf modifizierte oder neu erstellte Modelle der AixLib angewendet.**

***HTML\_<Branchname>:***

* Wird dieser Branch erstellt, hat die CI mögliche Fehler im HTML Code der Modelle entdeckt. Die CI korrigiert den HTML Code und pusht diesen auf den Branch. Die Änderungen sind in den Artifacts und können beim Erstellen Pull Request verglichen werden.

# Hilftsmittel

Folgende „commit messages“ triggern die CI und erledigt unterschiedliche Jobs. Die Erzeugnisse werden in einer Pipeline als Artifacts abgepeichert.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Commit | Wirkung | Artifacts |
| Trigger CI - give different reference results | Vergleicht die Referenzresult der AixLib und IBPSA. | * Unterschiedliche Referenzdateien |
| Trigger CI - Update reference results | Aktualisiert oder erstellt neue Referenzdateien, falls ein passendes .mos Skript existiert. | * Aktualisierte oder neu erstelle Referenzdatein |
| Trigger CI - Update WhiteList | Aktualisiert die WhiteList für die CI Tests | * Aktualisierte Whitelist |
| Trigger CI – Plot Model | Plottet die Ergebnisse eines Models aus dem RegressionTest als HTML Seite | * Google Chart Plot |
| Trigger CI – Only Regressiontest | Führt nur die RegressionTests aus | * Simulations.log * Error.log |
| Trigger CI – Only Check | Führt nur die Checks der Modelle durch | * Error.log |
| Trigger CI – Only Simulate | Simuliert der Modelle durch | * Error.log |

Weitere Informationen befinden sich unter diesen Links:

* <https://github.com/RWTH-EBC/AixLib/tree/development/bin>
* <https://github.com/RWTH-EBC/AixLib/wiki/AixLib-CI-GitLab>
* <https://www.ebc.eonerc.rwth-aachen.de/cms/E-ON-ERC-EBC/Das-Institut/Aktuelle-Meldungen-Institut/~llfhz/Qualitaetssicherung-der-Open-Source-Bibl/?lidx=1>