Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

|  |
| --- |
| Институт космических и информационных технологий |
| *институт* |
| Системы искусственного интеллекта |
| *кафедра* |

ОТЧЕТ О ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ

|  |
| --- |
| Кафедра «Системы искусственного интеллекта» ИКИТ СФУ |
| *Место прохождения практики* |
| Обзор литературы на тему связанности в объектно-ориентированных системах |
| *тема* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Руководитель от университета | | |  |  |  |  |
| Руководитель от предприятия | | |  |  |  |  |
|  | |  |  | *подпись, дата* |  | *инициалы, фамилия* |
| Студент | КИ17-11Б 031723019 | |  |  |  | В.А. Рудт |
|  | *номер группы, зачетной книжки* | |  | *подпись, дата* |  | *инициалы, фамилия* |

Красноярск 2019

TODO(Оглавление)

ВВЕДЕНИЕ

В данной работе рассматривается литература, представляющая с различных сторон тему измерения связанности (Coupling) в объектно-ориентированных системах.

Цель работы: с использованием имеющейся литературы рассмотреть, систематизировать и проанализировать текущие знания об измерении связанности в объектно-ориентированных системах.

Задачи:

* Изучить имеющеюся литературу на тему измерения связанности в объектно-ориентированных системах.
* Представить общий обзор имеющихся знаний на тему измерения связанности в объектно-ориентированных системах и проанализировать их.
* Определить перспективы развития исследований в области измерения связанности в объектно-ориентированных системах.

В последнее время, качество программного обеспечения (ПО) является важным аспектом. Однако, как можно определить это качество? При рассмотрении качества ПО нужно понять, с какой позиции можно его можно рассмотреть. С позиции пользователя, качество можно рассматривать как степень подверженности программы ошибкам, когда разработчик может воспринимать данное качество иначе. Для разработчика важной частью качественного ПО является его возможность и удобство расширяемости, а также его понятность. Если ПО не содержит данных качеств, то, даже если конкретная версия программы не содержит ошибок, новые ошибки неминуемо будут появляться.

В современном мире большую роль играет объектно-ориентированное программирование, так как данная парадигма легко воспринимается человеком.

Еще в начале развития объектно-ориентированных систем было определенно несколько метрик для их исследования. Одной из самых важных метрик является связанность. Данная метрика определяет степень взаимодействия между программными модулями. Многие исследователи считают, что в зависимости от степени связности будет определяться качество ПО. Также, некоторые из исследователей предлагают методы для измерения связанности. Различным методам измерения данной метрики, а также их анализу, посвящена данная работа.

1. Понятие и определение связанности

Дать четкое определение связанности достаточно проблематично, так как его понятие расплывчато. Однако, большинство понимают данный термин как силу связи или степень взаимодействия программных модулей между собой. В объектно-ориентированных системах такими модулями являются классы и объекты.

Неясность данного понятия состоит в том, что до конца не понятен весь спектр данного взаимодействия. Так же неясно, какую степень считать большой независимо от контекста. Последний вопрос является одним из ключевых в данной работе.

Теперь, когда было дано некоторое понятие о связанности, следует предоставить существующие попытки дать определение этому термину. Существует несколько определений, которые были получены в результате совместной работы институтов стандартизации ISO и IEC.

**Coupling**: the strength of the relationships between modules [1].

**Coupling**: manner and degree of interdependence between software modules [1].

**Coupling**: measure of how closely connected two routines or modules are [1].

**Coupling**: a measure of the interdependence among modules in a computer program [1].

Данные определения очень схожи между собой, однако, стоит дать собственное определение, которое будет использоваться в рамках данной работы. Также следует учитывать, что термин "coupling" пока не имеет четкого аналога на русском языке, так что в данной работе аналогом, как можно было заметить, является термин "связанность".

**Связанность:** степень взаимодействия между компонентами объектно-ориентированной системы (классами и объектами).

Данное определение очень узкое и подходит только для данного исследования, так как отражает два аспекта: связанность является степенью взаимодействия, и эта мера относится к объектно-ориентированным системам.

1. Измерение связанности

Измерением связанности, в основном, интересовались инженеры во время девяностых и нулевых годов. Сейчас выпускается относительно мало литературы на данную тему, что является негативным фактом. Нет литературы, которая бы некоторым образом подводила итоги об измерении связанности и представляла бы общую картину в целом.

Однако работы велись и были достигнуты некоторые успехи в этом направлении. Именно о них и пойдет речь в следующих пунктах.

* 1. Общие положения о связанности

Одно из первых упоминаний связанности встречается в книге Structured Design [2]. В данной книге связанность трактуется следующим образом:

Coupling is the measure of the strength of association established by a connection from one module to another [2].

Данное определение схоже с одним из тех, что мы упомянули выше. Имеется ввиду то определение, которое характеризует связанность как силу отношений между модулями.

Данная книга не уделяет времени вопросам на тему измерения связанности. К тому же, книга посвящена организации систем в функциональной и процедурной парадигме, а не в объектно-ориентированной. Однако, хотя в книге и не уделяется внимания непосредственно измерению связанности в объектно-ориентированных системах, в ней описаны некоторые закономерности, которые негативно влияют на связанность. А так как объектно-ориентированное программирование в какой-то степени выросло из функционального и процедурного, значит, данные закономерности будут работать и для этой парадигмы. Разберем их подробнее, так как они дают общее представление о том, каким правилам должна подчиняться связанность, чтобы её можно было оценить как хорошую.

Первая закономерность такова, что уменьшение количества связей между программными модулями уменьшает и количество путей, по которым изменения или ошибки могут распространиться на другие части системы [2].

Закономерность крайне важна, так как ошибки и изменения, переданные по связи другому модулю, могут перейти и к следующим модулям тоже и не факт, что следующий модуль будет один. Таким образом, изменения и ошибки нарастают волнообразно, как это проиллюстрировано на Рисунке 1.

Также, увеличение связей приводит к росту сложности системы, а значит, к сложности её поддержки и понимания [2].

Чтобы проиллюстрировать эту закономерность, автор приводит следующую ситуацию. Допустим, наша система имеет некую общую область данных, к которой обращаются модули, тем самым входя в некоторую общую среду. В таком случае, сами модули также взаимодействуют между собой, и включение нового модуля означает, что он будет включен в это общее окружение.



Рисунок 1— Волнообразный эффект изменений и ошибок

Однако, в таком случае, все имеют доступ к общей памяти, но не каждому следует её изменять. Изменение одним модулем общей памяти может вызвать ошибки в другом.

Отойдем от абстракций и немного конкретизируем ситуацию. Допустим, в системе имеется три модуля: модуль ввода, модуль вычислений и модуль вывода. Изобразим это на Рисунке 2.

На общую память влияют все три модуля, однако, если мы хотим ввести данные, вычислить новый результат и вывести его с исходными данными. Однако программист может случайно или намеренно изменить исходные данные еще на этапе вычислений, что приведет к некорректной работе программы.



Рисунок 2—Пример общей среды

Ситуация может существенно улучшиться, если доверить работу с памятью и её распределение только одному модулю так, как это показано на Рисунке 3.



Рисунок 3— Система без общей среды

Как можно заметить, число связей уменьшилось, что сделало нашу систему более помехоустойчивой, так как на память напрямую влияет только модуль контроля.

Таким образом, мы подходим к одному из главных принципов проектирования систем, однако для нас важна только одна его часть.

Принцы звучит так: “Low coupling, high cohesion”. Нас же интересует только первая его часть. После описанного выше становится понятно, что надежность системы определяется уровнем связанности. Чем ниже связанность, тем выше надежность системы. Это также относится и к объектно-ориентированным системам.

В книге “Structured design” автор W. P. Stevens определил некоторые факторы, которые влияют на уровень связанности. Факторы следующие: уровень сложности соединения модулей, ссылается ли модуль на другие модули или на себя, что именно отправляется и принимается между модулями.

Данные факторы очень расплывчато описывают связанность и не годятся для четкого измерения. Однако, с течением времени начали появляться некоторые более совершенные методы измерения связности, речь о которых пойдет далее.