

Опубликовано в материалах Международной научно-технической конференции «Информационно-математические технологии в экономике, технике и образовании». Екатеринбург: УГТУ–УПИ. 2007, с. 239–241.

Проблемы применения языков стандарта *IEC 61131-3* и возможные пути решения

В. А. Татарчевский

Уральский государственный горный университет

К концу 1980-х годов на рынке присутствовали сотни инструментальных средств программирования программируемых логических контроллеров (ПЛК). Задачу стандартизации средств программирования ПЛК взяла на себя Международная Электротехническая Комиссия (IEC). Результатом этой работы стала разработка стандарта *IEC 61131-3*. Однако цели создания стандарта (предоставление разработчикам унифицированного средства программирования ПЛК) в полном объеме были не достигнуты. Объединение в рамках одного стандарта пяти различных языков дезориентирует пользователей и затрудняет применение стандарта. Ниже рассмотрены языки стандарта и отмечены их основные недостатки.

LD (Ladder) является графическим языком, программа на котором представляет собой аналог релейно-контактной схемы. Однако такой подход вряд ли является целесообразным. Язык может быть использован только для описания процессов, имеющих дискретный характер. При обработке непрерывных переменных релейная аналогия теряет смысл. Диаграммы на этом языке сложны в отладке в силу своей неструктурированной природы.

FBD (Functional Block Diagram) является графическим функциональным языком программирования, использующим аналогию с цифровыми схемами. Программа на этом языке представляет собой совокупность функциональных блоков, соединенных линиями связи. Язык **FBD** является наиболее часто используемым языком стандарта *IEC*. Хотя этот язык обеспечивает обработку, как непрерывных сигналов, так и логических функций, в нем неудобным образом реализуются те участки программы, которые было бы удобно представить в виде конечного автомата.

Язык **SFC** (Sequential Function Chart) является графическим языком, в котором программа описывается в виде последовательности шагов, объединенных переходами. На этом языке естественным образом описываются технологические процессы, состоящие из последовательных шагов, с возможностью описания параллелизма. Для этого в языке используются средства разветвления и слияния потоков. Однако, при слиянии потоков возможно возникновение ряда трудноустраняемых ошибок. Язык допускает переход из шага, находящегося внутри «скобок» слияния и разветвления наружу. При этом программа теряет всякую осмысленность действий. Программная модель языка соответствует ординарной сети Петри.

Поэтому при недостаточном контроле могут оказаться одновременно активными несколько шагов, которые не находятся в параллельных потоках.

Язык *ST* (Structured Text) представляет собой текстовый язык высокого уровня. Этот язык может быть освоен технологом за короткий срок, однако текстовая форма представления программ – сдерживающий фактор при разработке сложных систем, так как не дает наглядного представления ни о структуре программы, ни о происходящих в ней процессах. Если для профессионального программиста такой порядок вещей представляется естественным, то, во всяком случае, привычным, то для программиста-технолога текстовая форма представления может служить серьезным препятствием.

Язык *IL* (Instruction List), являясь ассемблером виртуальной машины, имеет все те же недостатки, которые присущи ассемблерам реальных процессоров: сложность и высокую трудоемкость программирования, трудность модификации программ. Однако при этом он лишен главного достоинства ассемблера: высокой скорости выполнения программы, так как программа языке *IL* выполняется на виртуальном процессоре с довольно сложной структурой.

Возможно создание языка программирования ПЛК, который не имеет указанных выше недостатков.

Наиболее перспективным представляется применение в качестве базовой программной модели языка системы взаимодействующих конечных автоматов (КА). Такая модель наиболее естественным образом соответствует целевому назначению языка – описанию систем управления техническими объектами в силу того, что КА, как и большинство объектов управления имеет множество состояний, для которых в явном виде определены выходные воздействия и функции переходов. Система взаимодействующих КА (гиперавтомат) естественным образом включает в себя параллелизм, необходимый для описания поведения систем управления. Разделение программы на иерархические уровни, подобные модулям в универсальных языках высокого уровня, может быть обеспечено путем вложенности автоматов, а взаимодействие автоматов наиболее простым образом осуществляется посредством обмена сообщениями. Достоинствами такого языка программирования являются:

- единообразии описательных средств – вместо пяти языков *IEC 61131-3* используется один язык, простой в изучении и применении;
- широкие возможности моделирования систем. На этапе отладки ПО системы управления возможно описать модель объекта управления на этом же языке и производить отладку системы, учитывая взаимодействие с этой моделью;
- возможность графического представления программ. Конечные автоматы естественным образом представляются в графическом виде, что делает весьма удобным документирование ПО, его согласование с заказчиком и его анализ.

- Модель КА [1], которая лежит в основе языка, проста для понимания лицами, не являющимися профессиональными программистами, но хорошо владеющими предметной областью системы автоматизации (например, технологами). Это позволяет в ряде случаев производить разработку и модификацию ПО без привлечения профессиональных программистов.
- Иерархическое разбиение программы по уровням и использование обмена сообщениями [2, 3] вместо прямого вызова функций, характерного для процедурных языков высокого уровня, делает программные модули, обладающими высокой степенью автономности. Это позволяет проводить их разработку и отладку независимо.

Источники

1. *Шалыто А. А.* Switch-технология. Алгоритмизация и программирование задач логического управления. СПб.: Наука, 1998.
<http://is.ifmo.ru/books/switch/1>
2. *Гуисов М. И., Кузнецов А. Б., Шалыто А. А.* Интеграция механизма обмена сообщениями в Switch-технологию. СПбГУ ИТМО, 2003.
<http://is.ifmo.ru/projects/memech/>
3. *Альшевский Ю. А., Раер М. Г., Шалыто А. А.* Механизм обмена сообщениями для параллельно работающих автоматов (на примере системы управления турникетом). СПбГУ ИТМО, 2003.
<http://is.ifmo.ru/projects/turn/>