УДК 004.4'2:62-529

А.В. Пичкалев, начальник сектора, al-mail@iss-reshetnev.ru
АО «Информационные спутниковые системы» им. акад. М.Ф. Решетнева»,
г. Железногорск, Россия

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ РОБОТОТЕХНИЧЕСКИХ КОМПЛЕКСОВ С ИСПОЛЬЗОВАНИЕМ ГРАФИЧЕСКОГО ЯЗЫКА ДРАКОН

Аннотация. Статья описывает проблемы разработки управляющего программного обеспечения роботов. Предлагается решение проблем, основанное на внедрении в процесс разработки программного обеспечения гибридных языков программирования типа графического языка ДРАКОН.

Ключевые слова: графическое программирование, робототехнический комплекс, гибридный язык программирования, ДРАКОН

WORKING OUT OF THE SOFTWARE OF ONBOARD EQUIPMENT WITH USE GRAPHIC PROGRAMMING LANGUAGES

A.V. Pichkalev, head of sector, al-mail@iss-reshetnev.ru
JSC “ISS” ac. M. F. Reshetnev”, Zheleznogorsk, Russia

Abstract. Article describes problems of working out of the operating software of robots. Graphic programming gives the chance essentially other working out of the software. The decision of problems based on introduction in process of working out of the software of hybrid programming languages of type graphic language DRAKON is offered.

Key words: graphic programming, robots complex, a hybrid programming language, DRAKON

На сегодняшний момент робототехнические комплексы (РК) нашли применение только в производствах с высокой нормой прибыли и очень высокими капитальными затратами. Связано это в первую очередь с их высокой стоимостью и большим временем разработки и отладки. Перенастройка существующего РК под новые задачи вообще малоцелесообразна, т. к. дешевле разработать новый комплекс, чем разобраться в особенностях функциониро-

вания и возможностях применения прежнего. Основным компонентом высокой себестоимости РК является его управляющее программное обеспечение (ПО).

Процесс разработки ПО РК специфическими особенностями, отличными от разработки чисто программного продукта. Высокие требования к функциональным возможностям современных РК обуславливают высокую функциональную сложность применяемого ПО. В свою очередь, высокая функциональная насыщенность ПО требует значительных трудовых и временных затрат на разработку алгоритмов. В итоге себестоимость ПО начинает превышать стоимость аппаратных средств, а общая стоимость робототехнического комплекса промышленного применения делает его окупаемым только при массовом производстве (или при высокой конечной стоимости продукта, как, например, в атомной или космической промышленности). Таким образом, необходимо резко снизить стоимость и время разработки и отладки ПО робототехнических комплексов, чтобы сделать их применение массовым и относительно дешевым.

Традиционный подход при создании управляющего ПО заключается в описании исходных данных (алгоритмов функционирования и директив управления) в виде текстовых описаний, схем, таблиц и графиков и передачи этих материалов программистам. Программист должен учитывать особенности функционирования и конструкции РК, в которых он не является специалистом. Для облегчения задачи разработки ПО разрабатываются специальные исходные данные (ИД), где все это подробно описывается. Такой подход таит в себе значительный недостаток – возможность неверного толкования ИД программистом. Вероятность ошибки неверного толкования растет вместе с объемом ИД.

Возникает дополнительный участник разработки РК – алгоритмист – который должен обеспечивать взаимодействие между потребителями РК, разработчиками его конструкции и программистами, сводя воедино их разнородные требования и решая создаваемые этим проблемы (при этом он еще и должен решать собственную проблему корректной интерпретации требований к РК в ИД на разработку ПО). Зачастую разработчики алгоритмов функционирования РК не являются программистами по образованию, что также вносит лепту в качество ИД.

Для уменьшения вероятности возникновения ошибки применяются различные административные механизмы контроля над ошибками, которые в общем случае представляют собой некий фильтр, который, как любой фильтр, приводит к фазовому запаздыванию. Очевидно, внедрение дополнительных административных механизмов между разработчиком ИД и программистом приводит к задержке результатов разработки, притом на каждой итерации коррекции исходных данных. Снижается производительность, растут издержки.

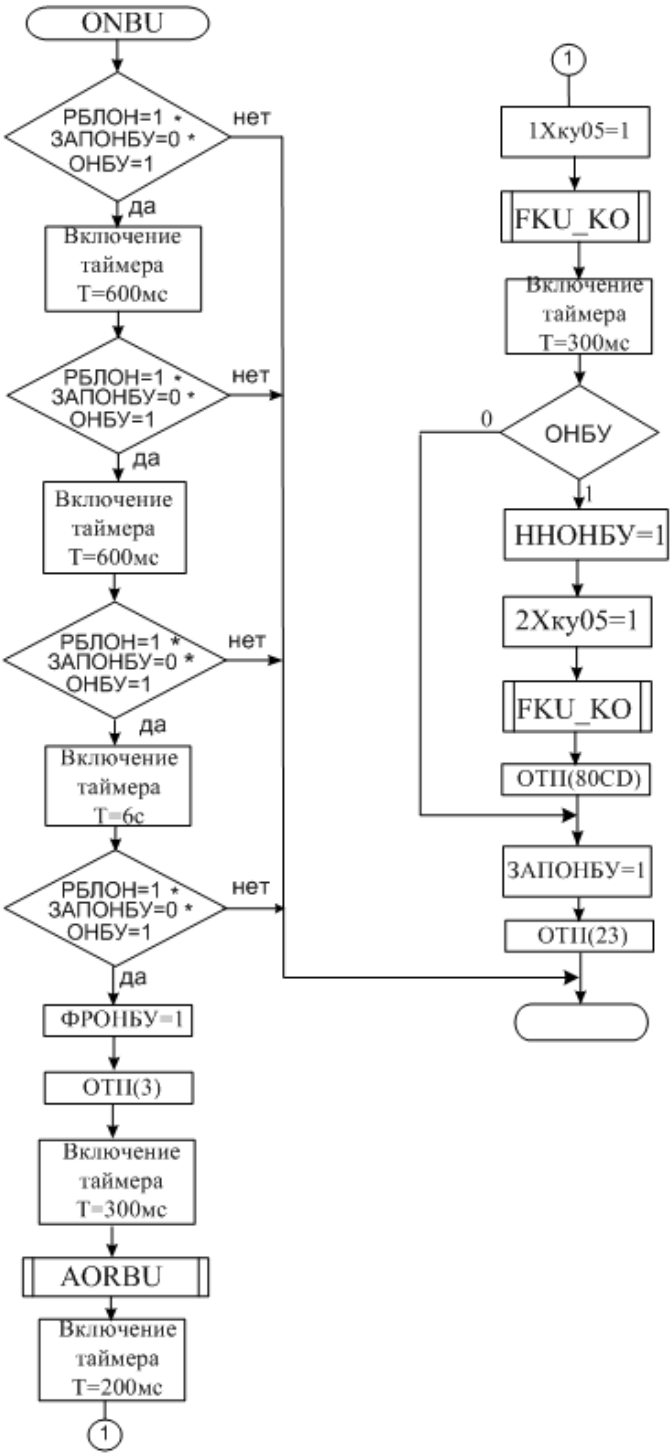

Рис. 1. Пример блок-схемы алгоритма управляющего ПО РК

Решить проблему можно было бы, если бы алгоритмист сразу создавал ПО РК. Но для этого он должен уметь одновременно программировать вычислитель РК и разбираться в его конструкции и особенностях функционирования, что требует от инженера незаурядной квалификации. Необходимо обеспечить алгоритмисту возможность создавать ПО РК, не вникая в тонкости программирования его вычислителя.

Это возможно, если графическую блок-схему алгоритма (которая обязательно приводится в ИД на ПО) сразу преобразовывать в исходный файл программы, исполняемой вычислителем РК. Такой способ разработки ПО называется графическим программированием и используется в соответствующих приложениях. Таких как LabVIEW, Simulink/MATLAB, Xcos/Scilab и других.

Решение проблемы выразилось еще в ходе реализации программы «Буран» в разработке специального графического языка программирования и моделирования ДРАКОН [1].

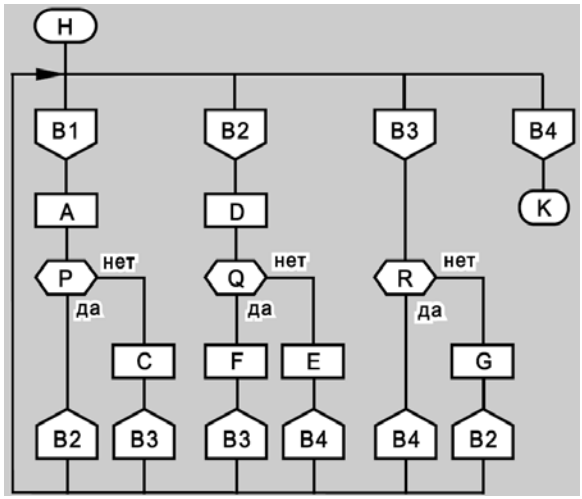

Рис. 2. Пример блок-диаграммы на языке ДРАКОН

Язык ДРАКОН основан на применении графических лексем, размещаемых по определенным правилам и соединяемых линиями связи [2]. Алгоритм, описанный на языке ДРАКОН, называют дракон-схемой. Лексемы организуются по так называемому «шампур-методу» вертикально в виде наборов, называемых ветками или «шампур-блоками». Ветки позволяют разделить алгоритм на смысловые части. То есть, дракон-схема является своего рода блок-схемой алгоритма ПО, реализованной в графическом дракон-редакторе (рис. 3).

В настоящее время для языка ДРАКОН разработаны графические редакторы (среды разработки), основанные на идее гибридных языков программирования и позволяющие синтезировать исходный код программных модулей на языках программирования высокого уровня. Реализация задаваемых в техническом задании команд управления бортовой аппаратурой в виде таких программных модулей позволит дальнейшую разработку ПО вести в среде графического программирования самому алгоритмисту без участия программиста.

Таким образом, использование языка ДРАКОН и гибридных языков может позволить полностью отказаться от традиционного подхода к разработке ПО РК, снижая интеллектуальную нагрузку на разработчика алгорит-

ма, исключая ошибки толкования исходных данных программистом. Использование гибридного языка программирования на основе языка ДРАКОН позволяет перейти к прямой компиляции программного кода из алгоритма. Это качество значительно повысит скорость разработки, позволит отказаться от административных механизмов контроля над ошибками. Но, самое главное – это позволит резко сократить затраты на разработку ПО РК, что сделает роботов более доступными для потребителей самого разного уровня.

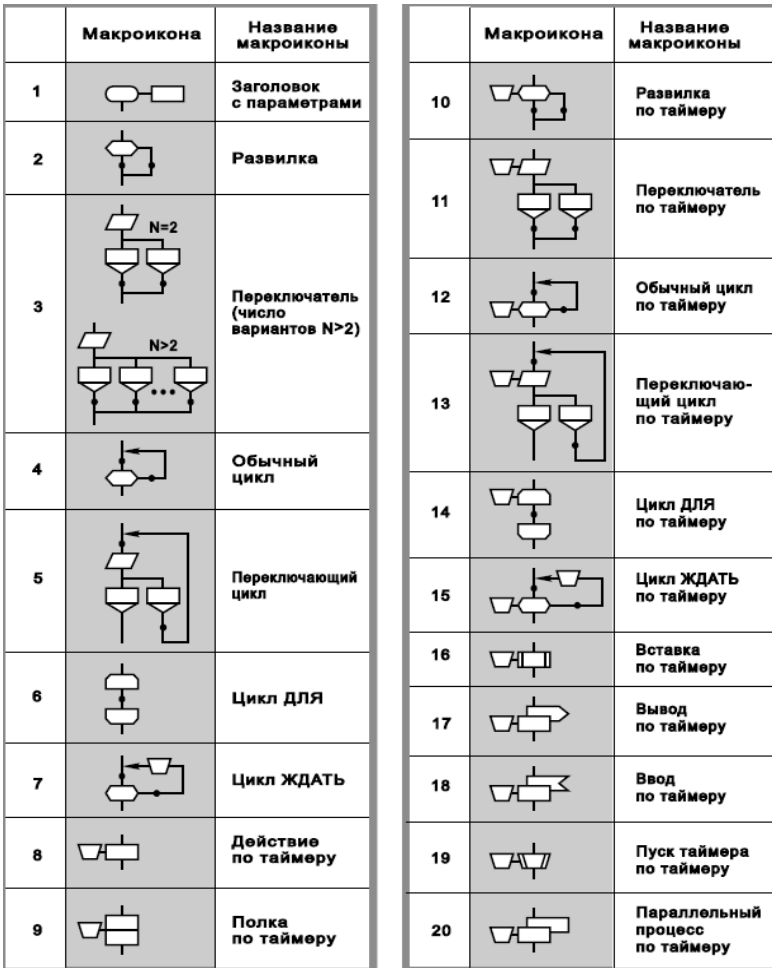

Рис. 3. Пример лексем языка ДРАКОН

Список литературы

1. Пароннджанов В.Д. Дружелюбные алгоритмы, понятные каждому. – М.:ДМК Пресс, 2010. – 463 с.

2. Пароннджанов В.Д. Как улучшить работу ума. Алгоритмы без программистов – это очень просто! – М.: Дело, 2001. – 360 с.