

Автоматизация анализа отработки циклограмм бортовых вычислителей космических аппаратов

Табакон Е.В.^{1*}, Зинина А.И.^{2}, Красавин Е.Э.^{2***}**

¹Московское опытно-конструкторское бюро «Марс»,

1-й Щемилонский пер., 16, стр. 2, Москва, 127473, Россия

²Московский авиационный институт (национальный исследовательский университет), МАИ, Волоколамское шоссе, 4, Москва, ГСП-3, 125993, Россия

**e-mail: e-tab@rambler.ru*

***e-mail: zinina0818@mail.ru*

****e-mail: krasavin.ieghor@mail.ru*

Статья поступила 26.03.2020

Аннотация

В работе рассматривается задача автоматизации анализа использования ресурсов вычислителей космических аппаратов. Рассмотрен существующий подход, выявлены его недостатки. Предложен алгоритм автоматического проведения подобного анализа. На основании алгоритма разработано программное средство, для обработки сведений о степени использования программными модулями ресурсов бортового вычислителя по данным телеметрической информации и нормативной документации. Разработанное программное средство позволяет автоматически сформировать данные для отчётной документации, что уменьшает количество времени, затрачиваемого на её составление.

Ключевые слова: система управления, бортовой вычислитель, телеметрическая информация, анализ, космический аппарат.

Введение

Создание космического аппарата и поддержание его работоспособности является весьма сложной задачей [1, 2, 3, 4]. Одной из частей космического аппарата является его система управления [5]. В процессе разработки и эксплуатации системы возникает вопрос об обеспечении её безотказной работы [6]. Крайне важным, при решении данного вопроса является анализ телеметрической информации. В ней содержатся сведения об общем состоянии аппарата и работе всех его компонентов, в том числе и бортового вычислителя [7, 8].

Бортовой вычислитель является частью системы управления космическим аппаратом [9, 10, 11] и содержит в своём составе несколько идентичных вычислительных устройств («граней»). Исполняемые в нём модули бортового программного обеспечения (ПО) решают основные задачи системы управления и обеспечивают информационное взаимодействие с другими подсистемами аппарата. [12, 13].

Система управления космическим аппаратом является, по своей сути, системой «жёсткого» реального времени. Время выполнения каждого программного модуля чётко регламентируется [14, 15]. Этот регламент представляет собой циклограмму работы бортового вычислителя. Циклограмма работы вычислителя содержится в файле. Выдвигается требование, чтобы времена работы модулей на бортовом вычислителе строго укладывались в отведённые рамки. Поэтому является весьма важным получение информации об этих временах работы.

Классический метод анализа длительности исполнения ПО заключается в составлении таблиц, где содержатся сведения о временах работы программных

модулей на гранях бортового вычислителя. Такие таблицы позволяют точно идентифицировать те модули, которые выходят за рамки установленного времени.

Однако составление этих таблиц требует большого количества времени, т.к. массивы с телеметрической информацией содержат большое количество различных данных, представленных в неудобном для человеческого восприятия виде. Кроме того, при производстве подобного рода документов существует большая вероятность совершения ошибок в силу невнимательности человека-составителя. Вследствие этого возникает задача автоматизации процесса составления таких таблиц и анализа нормативности длительностей исполнения программных модулей.

В рамках работы для решения поставленной выше задачи рассматривается разработка программного средства, которое в автоматическом режиме бы находило сведения в телеметрической информации о времени работы модулей бортового ПО в гранях вычислителя и составляло бы отчёт в виде таблицы с суждениями о нормативности функционирования.

Предлагаемый алгоритм

Исходными данными для разработки является информация о времени выполнения программных модулей бортового ПО. Эта информация содержится в файлах телеметрической информации, которые подаются на вход алгоритма. В процессе обработки происходит вычисление параметров таймирования и сравнение полученных значений с выделенным временем

Результатами обработки являются времена завершения работы программных модулей на гранях бортового вычислителя в миллисекундах согласно

телеметрической информации и согласно циклограмме, а также информация о запасе времени их работы, как в абсолютном значении, так и в процентах.

На следующем шаге полученные результаты записываются в файл таким образом, чтобы формировать таблицу, из которой удобно извлекать нужную информацию.

Исходные данные

Исходя из того, что программе для работы требуется файл с телеметрической информацией, файл с информацией циклограмм и имя используемой циклограммы необходимо, чтобы программа обеспечивала механизм приёма данных сведений от пользователя.

Телеметрическая информация о времени завершения работы программных модулей на гранях бортового вычислителя имеет строго определённый формат, который включает в себя:

- Время формирования данной информации;
- Идентификатор;
- Значение.

Количество работающих граней бортового вычислителя определяется специальной переменной, где каждый разряд определяет включённое/выключенное состояние соответствующей грани бортового вычислителя.

Циклограмма работы вычислителя формируется разработчиками бортового ПО и документируется. С целью обеспечения возможности автоматической обработки информации циклограммы, на основе документации разрабатывается

файл циклограммы. Формат представления сведений в данном файле представляет собой имя циклограммы, времена завершения работы программных модулей и их имена.

Описание представления результатов

Таблица с результатами должна представлять собой файл OpenOffice Calc (формат .ods). Данный формат был выбран в связи с тем, что он является свободно распространяемым, и существуют простые методы по его автоматизации [16]. Для улучшения восприятия информации из формируемой программой таблицы её соседние строки должны иметь разные цвета — как вариант, могут быть использованы два чередующихся цвета. Также необходимо обеспечить возможность быстрой идентификации строк таблицы, которые соответствуют программным модулям, для которых запас времени работы в процентах по отношению к значению, заданному по циклограмме, меньше некоторого порогового значения.

Для предотвращения ошибок работы бортового вычислителя из-за превышений времени выполнения команд необходимо отслеживать тенденцию изменений параметров таймирования в процессе работы КА. Исследования показали, что со временем показатели времени выполнения команд увеличиваются. Поэтому, необходимо выделять значения близкие к номинальным (значениям из документации). В настоящее время, критерием «опасной близости» параметра к номиналу является значение запаса (разность номинала и реального значения) менее 30%.

В результате автоматизированного анализа на нормативность времени исполнения программных модулей должно формироваться следующее информационное сообщение:

1. Имя файла протокола (в окне программы обозначен как выходной файл);
2. Имя циклограммы;
3. Количество вызова программных модулей в циклограмме;
4. Количество точек в телеметрической информации;
5. Количество точек с запасом <30%.

Графический интерфейс

С целью упрощения использования программа должна иметь простой и удобный для пользователя графический интерфейс. Также в соответствии с пожеланиями к разработчикам требуется, чтобы программа запоминала положение своего окна и открывала своё окно в том положении, в котором оно было закрыто прошлый раз.

Программная реализация

Предложенный алгоритм был реализован в виде программного приложения «Ziklogramma».

Как правило, программное обеспечение, используемое на предприятии для разработки и проверки систем управления, написано на языке C/C++ [17]. Поэтому, чтобы обеспечить совместимость разрабатываемого программного средства с уже существующим программным обеспечением, языком разработки выбран C++.

Разработка программы велась в среде QtCreator, позволяющей создавать программное обеспечение с графическим интерфейсом. Выбор данной среды обусловлен её широкой распространённостью на предприятии.

Для создания *.ods файла был применён способ, при котором с помощью программы архиватора создаётся архив данного формата.

Все полезные данные в документе формата *.ods содержатся в файле content.xml. Это позволяет создавать этот xml-файл с данными, которые должны содержаться в готовом документе и затем записывать его в нужный архив с помощью архиватора.

Для корректного открытия файла *.ods, как электронной таблицы, средствами программы OpenOffice, данный архив должен также содержать папку META-INF с файлом manifest.xml. Содержимое этого файла является стандартным и не зависит от данных, хранящихся в документе *.ods.

Выбранный способ работы с подобными документами исключает необходимость покупки каких-либо библиотек. Созданная программа для создания архива формата *.ods и записи в него необходимых файлов и папок использует бесплатный архиватор 7zip для командной строки — приложение 7za.exe.

Для создания и работы с xml файлами используется популярная библиотека TinyXml.

Современные тенденции разработки прикладного программного обеспечения предполагают использование системы контроля версий, которая позволяет облегчить работу с разными версиями программного продукта, что упрощает

процесс разработки [18, 19, 20]. В процессе разработки данной программы использована типовая система контроля версий Git.

Описание работы программы

Вид главного окна программы показан на рисунке 1.

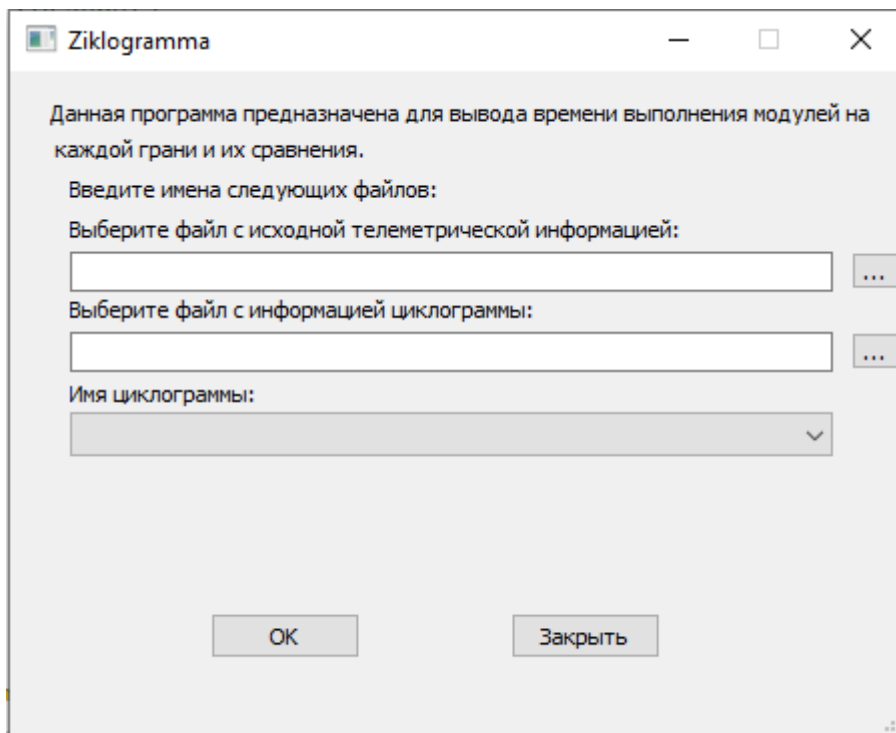


Рисунок 1 Главное окно программы

В соответствии с описаниями полей пользователь должен выбрать или указать следующие данные:

- Имя файла с исходной телеметрической информацией (или полный путь к данному файлу);
- Имя файла с информацией циклограммы (или полный путь к данному файлу);
- Имя циклограммы.

После заполнения всех полей пользователь должен нажать кнопку «ОК» (см. рисунок 1).

Если введённые данные верны, и программа отработала успешно, то по завершении формирования файла отчёта отрывается окно, вид которого показан на рисунке 2.

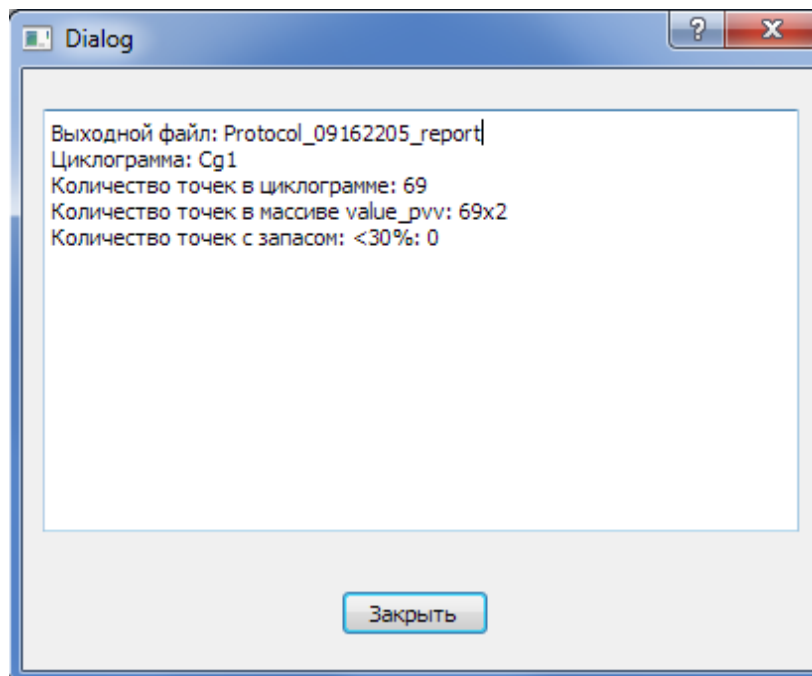


Рисунок 2 Окно со справочной информацией

На рисунке 2 показано, что после успешного выполнения программа выводит указанную в требованиях справочную информацию:

Результатом работы является файл формата *.ods, который представляет собой электронную таблицу. Пример такой таблицы представлен на рисунке 3.

Number of point	Name of module	First side	Second side	Max value	Actual time of the work of the side	Ending time in CG	Time in CG	Stock	Stock, %
1	SyncBuk_x	1080	470	1080	780.000000	2000	2000	1220	61.000000
2	MkoExRecv	2880	2730	2880	880.000000	3400	1400	520	37.142857
3	MkoSend_zp2	3820	3490	3820	420.000000	4700	1300	880	67.692307
4	disp_skd_x	5500	5480	5500	800.000000	6000	1300	500	38.461540
5	MkoRecv_zp2	6340	6490	6490	490.000000	7250	1250	760	60.799999
6	MkoSend_zp3	8130	7340	8130	880.000000	8600	1350	470	34.814816
7	disp_skd_x	8880	8880	8880	280.000000	9900	1300	1020	78.461540
8	MkoRecv_zp3	10800	10780	10800	900.000000	11500	1600	700	43.750000
9	MkoSend_zp4	12070	11590	12070	570.000000	13000	1500	930	62.000000
10	disp_skd_x	13480	13470	13480	480.000000	14600	1600	1120	70.000000
11	MkoRecv_zp4	15130	15380	15380	780.000000	16400	1800	1020	56.666668
12	disp_skd_x	16980	16980	16980	580.000000	17650	1250	670	53.599998
13	SendPw_x	18230	17750	18230	580.000000	18850	1200	620	51.666668
14	RecvPw_x	18980	19450	19450	600.000000	20100	1250	650	52.000000
15	disp_skd_x	20910	21240	21240	1140.000000	21750	1650	510	30.909090
16	SendPw_x	22000	22000	22000	250.000000	22200	450	200	44.444443
17	RecvPw_x	22630	22630	22630	430.000000	22950	750	320	42.666668
18	disp_skd_x	24290	24290	24290	1340.000000	25000	2050	710	34.634148
19	MkoSend_zp5	25650	25090	25650	650.000000	27000	2000	1350	67.500000
20	disp_skd_x	27120	27120	27120	120.000000	28800	1800	1680	93.333336
21	SendPw_x	28900	28900	28900	100.000000	31000	2200	2100	95.454544
22	MkoRecv_zp5	31670	31930	31930	930.000000	33500	2500	1570	62.799999
23	psl1_el3_x	33670	33670	33670	170.000000	33900	400	230	57.500000
24	MkoSend_zp6	34620	33990	34620	720.000000	35900	2000	1280	64.000000
25	Empty	0	0	0	0.000000	37400	1500	1500	100.000000
26	disp_skd_x	37680	37680	37680	280.000000	38900	1500	1220	81.333336
27	MkoRecv_zp6	39710	40000	40000	1100.000000	41000	2100	1000	47.619049
28	disp_skd_x	41590	41590	41590	590.000000	42000	1000	410	41.000000
29	SendPw_x	42410	42110	42410	410.000000	42800	800	390	48.750000
30	RecvPw_x	42930	43230	43230	430.000000	43600	800	370	46.250000
31	disp_skd_x	44170	44400	44400	800.000000	44800	1200	400	33.333332
32	SendPw_x	44970	44970	44970	170.000000	45300	500	330	66.000000
33	RecvPw_x	45560	45560	45560	260.000000	45800	500	240	48.000000
34	disp_skd_x	46610	46610	46610	810.000000	49000	3200	2390	74.687500

Рисунок 3 Таблица протокола

На этом рисунке показано, что строки таблицы обладают чередующимися цветами. Данная форма представления информации призвана облегчить восприятие. А в случае, если значение запаса в процентах в крайне правом столбце таблицы, для какого-то из программных модулей принимает значение меньше порогового, то соответствующая строка получает окраску красного цвета. На рисунке 4 показан пример таблицы для случая, когда пороговое значение установлено на уровне 30%. Оно может быть легко изменено правкой в коде программы.

Number of point	Name of module	First side	Second side	Max value	Actual time of the work of the side	Ending time in CG	Time in CG	Stock	Stock, %
1	SyncBuk_x	1080	470	1080	780.000000	2000	2000	1220	61.000000
2	MkoExRecv	2880	2730	2880	880.000000	3400	1400	520	37.142857
3	MkoSend_zp2	3820	3490	3820	420.000000	4700	1300	880	67.692307
4	disp_skd_x	5500	5480	5500	800.000000	6000	1300	500	38.461540
5	MkoRecv_zp2	6340	6490	6490	490.000000	7250	1250	760	60.799999
6	MkoSend_zp3	8130	7340	8130	880.000000	8600	1350	470	34.814816
7	disp_skd_x	8880	8880	8880	280.000000	9900	1300	1020	78.461540
8	MkoRecv_zp3	10800	10780	10800	900.000000	11500	1600	700	43.750000
9	MkoSend_zp4	12070	11590	12070	570.000000	13000	1500	930	62.000000
10	disp_skd_x	13480	13470	13480	480.000000	14600	1600	1120	70.000000
11	MkoRecv_zp4	15130	15380	15380	780.000000	16400	1800	1020	56.666668
12	disp_skd_x	16980	16980	16980	580.000000	17650	1250	670	53.599998
13	SendPvv_x	18230	17750	18230	580.000000	18850	1200	620	51.666668
14	RecvPvv_x	18980	19450	19450	600.000000	20100	1250	650	52.000000
15	disp_skd_x	20910	21240	21240	1140.000000	21750	1650	510	30.909090
16	SendPvv_x	22000	22000	22000	250.000000	22200	450	200	44.444443
17	RecvPvv_x	22630	22630	22630	430.000000	22950	750	320	42.666668
18	disp_skd_x	24290	24290	24290	1340.000000	25000	2050	710	34.634148
19	MkoSend_zp5	25650	25090	25650	650.000000	26800	1800	1150	63.888889
20	disp_skd_x	27120	27120	27120	320.000000	28600	1800	1480	82.222221
21	SendPvv_x	28900	28900	28900	300.000000	30800	2200	1900	86.363640
22	MkoRecv_zp5	31670	31930	31930	1130.000000	32000	1200	70	5.833333
23	psl1_el3_x	33670	33670	33670	1670.000000	32300	300	-1370	-456.666656
24	MkoSend_zp20	34620	33990	34620	2320.000000	32900	600	-1720	-286.666656
25	Empty	0	0	0	0.000000	33500	600	600	100.000000
26	MkoRecv_zp20	37680	37680	37680	4180.000000	34250	750	-3430	-457.333344
27	psl1_el3_x	39710	40000	40000	5750.000000	34450	200	-5550	-2775.000000
28	MkoSend_zp21	41590	41590	41590	7140.000000	35050	600	-6540	-1090.000000

Рисунок 4 Вид таблицы при наличии строк с недостаточным запасом времени

Заключение

В процессе работы решена задача по автоматизации анализа использования ресурсов бортовых вычислителей космических аппаратов и разработан инструмент, позволяющий легко и быстро создавать электронные таблицы с информацией о времени исполнения программных модулей в гранях бортового вычислителя, его запасе, как в абсолютном значении, так и в процентах, а также формировать суждения о нормативности данных показателей. Это позволяет оперативно получать информацию о соответствии фактических времён работы программных модулей максимально допустимым и тем самым увеличить надёжность работы ПО в вычислителе.

Направлением дальнейшей работы является обеспечение взаимодействия со средствами автоматизации испытаний бортового ПО, с целью автоматизации получения телеметрической информации, запуска обработки, записи результатов в базу данных предприятия и генерации отчётных документов.

Библиографический список

1. Алифанов О.М., Медведев А.А., Соколов В.П. Подходы к созданию и направления применения малых космических аппаратов в космической деятельности // Труды МАИ. 2011. № 49. URL: <http://trudymai.ru/published.php?ID=28039>
2. Гусев А.А., Ильина И.Ю., Усачев О.А. Разработка перспективной космической платформы для космических аппаратов океанографического назначения серии «Метеор» // Труды МАИ. 2014. № 74. URL: <http://trudymai.ru/published.php?ID=49285>
3. Асюшкин В.А., Викуленков В.П., Ишин С.В. Итоги создания и начальных этапов эксплуатации межорбитальных космических буксиров типа "Фрегат" // Вестник НПО им. С.А. Лавочкина. 2014. № 1. С. 3 - 9.
4. Внуков А.А., Рвачева Е.И. Предпосылки и перспективы создания полностью электрореактивных космических аппаратов для работы на геостационарной орбите // Сибирский журнал науки и технологий. 2014. № 4 (56). С. 140 - 146.
5. Матюшин М.М., Луценко Ю.С., Гершман К.Э. Синтез структуры органа управления космических группировок // Труды МАИ. 2016. № 89. URL: <http://trudymai.ru/published.php?ID=72869>
6. Заведеев А.И., Ковалев А.Ю. Диагностика состояния и принципы повышения отказоустойчивости бортовой системы управления космического аппарата // Труды МАИ. 2012. № 54. URL: <http://trudymai.ru/published.php?ID=29687>
7. Бровкин А.Г., Бурдыгов Б.Г., Гордийко С.В. и др. Бортовые системы управления космическими аппаратами. – М.: Изд-во МАИ-ПРИНТ, 2010. – 304 с.

8. Пейсахович Д.Г. Некоторые особенности построения систем передачи телеметрической информации // Молодой ученый. 2010. Т.1. № 8. С. 109 - 112.
9. T. Peng et al. A Component-Based Middleware for a Reliable Distributed and Reconfigurable Spacecraft Onboard Computer // IEEE 35th Symposium on Reliable Distributed Systems (SRDS), Budapest, 2016, pp. 337 – 342. DOI:[10.1109/SRDS.2016.051](https://doi.org/10.1109/SRDS.2016.051)
10. Eickhoff J. Onboard computers, onboard software and satellite operations: an introduction, Springer Science & Business Media, 2011. DOI 10.1007/978-3-642-25170-2.
11. Додонов А.Р. Принципы организации бортовых вычислительных комплексов автоматических космических аппаратов // Достижения науки и образования. 2018. № 8 (30). URL: <https://cyberleninka.ru/article/n/printsipy-organizatsii-bortovyh-vychislitelnyh-kompleksov-avtomaticheskikh-kosmicheskikh-apparatov>
12. Nancy G. Leveson. Role of software in spacecraft accidents // Journal of spacecraft and Rockets, 2004, vol. 41 (4), pp. 564 - 575.
13. Андреев В.П., Волович Н.В., Глебов В.М. и др. Проектирование и испытания бортовых систем управления. – М.: Изд-во МАИ-ПРИНТ, 2011. – 344 с.
14. Salehi Mohammad et al. Two-state checkpointing for energy-efficient fault tolerance in hard real-time systems // IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2016, vol. 24 (7), pp. 2426 - 2437. DOI: [10.1109/TVLSI.2015.2512839](https://doi.org/10.1109/TVLSI.2015.2512839)
15. Xu J. and Parnas D.L. On satisfying timing constraints in hard-real-time systems // IEEE transactions on software engineering, 1993, vol. 19 (1), pp. 70 - 84. DOI: 10.1109/32.210308

16. Benson, Calum, Matthias Muller-Prove, Jiri Mzourek. Professional usability in open source projects: GNOME, OpenOffice. org, NetBeans // CHI'04 extended abstracts on Human factors in computing systems, 2004, Vienna, Austria.
17. СИНИЦЫН С.В., Орлов Д.В. Системное бортовое программное обеспечение. Операционная среда разработки. – М.: МОКБ «МАРС», 2018. – Ч. 1. - 148 с.
18. John D. Blischak, Emily R. Davenport, and Greg Wilson. A quick introduction to, version control with Git and GitHub // PLoS computational biology, 2016, vol. 12 (1). DOI: 10.1371/journal.pcbi.1004668
19. Loeliger Jon, Matthew McCullough. Version Control with Git: Powerful tools and techniques for collaborative software development, O'Reilly Medvia, Inc., USA, 2012, 400 p.
20. Pilato C.M., Collins-Sussman B., Fitzpatrick B.W. Version control with subversion: next generation open source version control, O'Reilly Media, Inc., 2008, 432 p.