

Legacy Information Systems: Issues and Directions

Jörg Hodapp

Hochschule Furtwangen, Application Architectures Master - Semester 1

28. Mai 2009

Inhaltsverzeichnis

- 1 Kernaussagen des Papers
- 2 Meinung der Autoren
- 3 Stellungnahme
- 4 Literatur
- 5 Fragen/Diskussion

Inhaltsverzeichnis

- 1 Kernaussagen des Papers
- 2 Meinung der Autoren
- 3 Stellungnahme
- 4 Literatur
- 5 Fragen/Diskussion

Was sind Legacy Information Systems [LIS]?

Computersysteme und/oder Anwendungen mit folgenden Eigenschaften:

- sowohl langsame/veraltete Hardware als auch veraltete Technologie
- Mangel an Dokumentation und Kenntnissen (vgl. auch Masak (2009))
- Fehler: schwer und aufwändig zu analysieren/beheben
- keine klaren Schnittstellen (schwer zu integrieren/erweitern)
- hoher Wartungsaufwand (vgl. auch Masak (2009))
- für Unternehmenserfolg zwingend notwendig (vgl. auch Masak (2009))

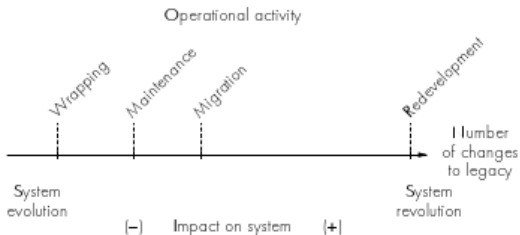
Was sind Legacy Information Systems [LIS]?

Computersysteme und/oder Anwendungen mit folgenden Eigenschaften:

- sowohl langsame/veraltete Hardware als auch veraltete Technologie
- Mangel an Dokumentation und Kenntnissen (vgl. auch Masak (2009))
- Fehler: schwer und aufwändig zu analysieren/beheben
- keine klaren Schnittstellen (schwer zu integrieren/erweitern)
- hoher Wartungsaufwand (vgl. auch Masak (2009))
- für Unternehmenserfolg zwingend notwendig (vgl. auch Masak (2009))

Bewältigungsstrategien

nach Bisbal et al. (1999):



Quelle: Bisbal et al. (1999), S. 3

Neuentwicklung und Wrapping

Neuentwicklung:

- neue Hardware-Plattform, moderne Architekturen und Datenbanken
- hohes Risiko (LIS sind umfangreich und komplex)
- Problem der Technologie- und Anforderungsänderungen

Wrapping:

- Kapselung in neu definierte Schnittstellen
- Spezielle Form: *screen scraping* (vgl. Masak (2006), S. 96, spricht hier von *Refronting*)
- jedoch weiterhin hoher Wartungsaufwand

Neuentwicklung und Wrapping

Neuentwicklung:

- neue Hardware-Plattform, moderne Architekturen und Datenbanken
- hohes Risiko (LIS sind umfangreich und komplex)
- Problem der Technologie- und Anforderungsänderungen

Wrapping:

- Kapselung in neu definierte Schnittstellen
- Spezielle Form: *screen scraping* (vgl. Masak (2006), S. 96, spricht hier von *Refronting*)
- jedoch weiterhin hoher Wartungsaufwand

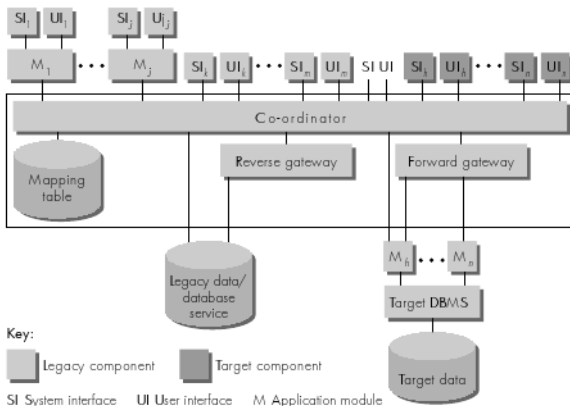
Migration

- Alternative zu risikoreichen Neuentwicklung und unzureichendes Wrapping
- individuell
- Umzug des LIS
 - neue Plattform
 - Beibehalt der Funktionalität
 - geringe Geschäftsunterbrechung
- umfasst zahlreiche Gebiete des SW-Engineerings
- Wiederverwendung von LIS-Komponenten; parallele Entwicklung und Inbetriebnahme
- Augenmerk auf Datenmigration (wichtige Geschäftsressource)

Chicken-Little-Strategy (I)

- elf inkrementelle Schritte (vgl. Bisbal et al. (1999), S. 8)
- Vermittlungsmodule zwischen Systemen
- paralleler Betrieb
- *Overhead* durch Vermittlungsmodule
- Features moderner DBMS unter Umständen nicht nutzbar

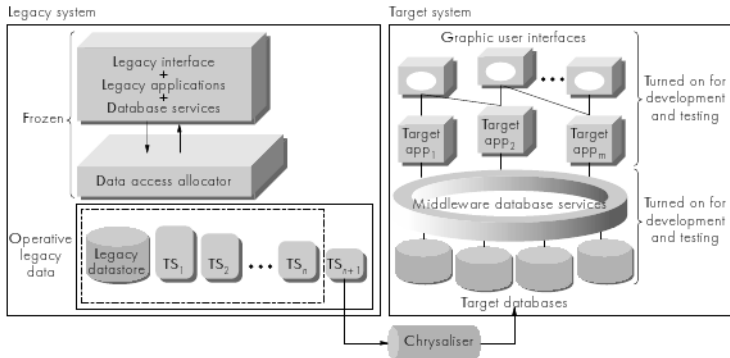
Chicken-Little-Strategy (II)



Quelle: Bisbal et al. (1999), S. 7

Butterfly-Methodik (I)

- LIS während der Migration betriebsfähig
- keine Vermittlungsmodule



Quelle: Bisbal et al. (1999), S. 8

Butterfly-Methodik (II)

- geringe Geschäftsunterbrechung (vgl. auch Erdle (2005), S. 13)
- vor Umschaltung auf neues System: umkehrbar/gefahrenlos unterbrechbar
- keine Interaktion zwischen den Systemen

Inhaltsverzeichnis

- 1 Kernaussagen des Papers
- 2 Meinung der Autoren**
- 3 Stellungnahme
- 4 Literatur
- 5 Fragen/Diskussion

Meinung der Autoren

- früher Stand der Kenntnisse
- weitere Untersuchung aller Migrationsaspekte
- Basis: Chicken-Little-Strategy und Butterfly-Methodik
- Untersuchungsobjekte:
 - Klassifizierung von LIS (spezifische Strategien)
 - Werkzeuge/Tools
- Anwendungsfälle (Erfahrungen aus Praxis)

Inhaltsverzeichnis

- 1 Kernaussagen des Papers
- 2 Meinung der Autoren
- 3 Stellungnahme**
- 4 Literatur
- 5 Fragen/Diskussion

Stellungnahme (I)

Was sind LIS?

- Kriterium (eigentlich) inakzeptabler Wartungsaufwand
- Alterung nicht durch Zeit sondern durch Umwelteinflüsse/Anforderungsänderungen bedingt

Bewältigungsstrategien

- heutige Wissenstand ähnlich
- Masak (2006): zusätzlich COTS-SW und weitere Aufteilung/Verfeinerung der Strategien

Stellungnahme (I)

Was sind LIS?

- Kriterium (eigentlich) inakzeptabler Wartungsaufwand
- Alterung nicht durch Zeit sondern durch Umwelteinflüsse/Anforderungsänderungen bedingt

Bewältigungsstrategien

- heutige Wissenstand ähnlich
- Masak (2006): zusätzlich COTS-SW und weitere Aufteilung/Verfeinerung der Strategien

Stellungnahme (II)

Migrationsmethoden: Chicken-Little-Strategy

- inkrementell: übersichtlich & testbar (vgl. Erdle (2005), S. 12)
- geringe Geschäftsunterbrechung (vgl. Erdle (2005), S. 13)
- paralleler Betrieb
- Zerlegung in Komponenten/Module
- Vermittlungsmodule: aufwändig und nicht trivial (vgl. Erdle (2005), S.13)
- hohe Last (vgl. Masak (2006), S. 128 f.) und geringe Performance

Stellungnahme (III)

Migrationsmethoden: Butterfly-Methodik

- Anzahl der temporärer Speicher hoch (nach Aktivität)
- *Data access allocator* und *Chrysaliser* aufwändig und nicht trivial (vgl. Erdle (2005), S. 13)
- Einsatz des Neusystems nach Abschluss der Migration (vgl. Masak (2006), S. 129)
- *Chrysaliser* muss schneller sein als Aufbau temporärer Speicher

Stellungnahme (IV)

Fazit

- keine allgemeinen Migrationsregeln/Lösungsstrategien, da
 - LIS und Zielsystem unternehmensindividuelle Systeme sind
 - Vielzahl von LIS
 - LIS: hochkomplex (muss analysiert und verstanden werden)
- Ansätze zur Migration unzureichend, da
 - viele (Vor- und) Nachteile
 - größter Nachteil ist Komplexität der Vermittlungs- bzw. Transformationsmodule
 - Vermittlungs-/Transformationsmodule schwer/unmöglich bei umfangreiche Änderungen in Datenhaltungsschema zu entwickeln
- immer noch Forschungs- und Studiengebiet

Inhaltsverzeichnis

- 1 Kernaussagen des Papers
- 2 Meinung der Autoren
- 3 Stellungnahme
- 4 Literatur**
- 5 Fragen/Diskussion

Literatur

- Bisbal, Jesus/Lawless, Deirdre/Wu, Bing et al. (1999): Legacy Information Systems: Issues and Directions, http://www.tecn.upf.es/~jbisbal/publications/IEEESoftware16_5.pdf (Zugriff am 05.05.2009)
- Masak, Dieter (2006): Legacysoftware, Das lange Leben der Altsysteme, Berlin: Springer
- Masak, Dieter (2009): Vortrag (15.05.2009) im Zuge der Vorlesung "Integration Engineering" des Studiengangs "Application Architectures Master" an der Hochschule Furtwangen
- Erdle, Christoph (2005): Legacy Migrationsstrategien, http://www4.in.tum.de/lehre/seminare/hs/WS0506/mvs/files/Ausarbeitung_Erdle.pdf (Zugriff am 20.05.2009)

Inhaltsverzeichnis

- 1 Kernaussagen des Papers
- 2 Meinung der Autoren
- 3 Stellungnahme
- 4 Literatur
- 5 Fragen/Diskussion

Fragen und Diskussion

