



Tema 2: Leyes de Lehman

Evolución y Adaptación de Software

Carlos E. Cuesta, ETSII, URJC



Universidad
Rey Juan Carlos





Leyes de Lehman

- Leyes de la Evolución de Software
- Formuladas por Meir M. (Manny) Lehman y László (Les) Belady
 - Por primera vez en 1974
 - Revisadas sucesivamente en 1978, 1980, 1991 y 1997
- Intentan describir el equilibrio entre las fuerzas que impulsan nuevos desarrollos (esto es, impulsan el cambio) y las que, por el contrario, frenan el progreso (es decir, producen resistencia)
 - Describen *ciclos de realimentación (feedback)*
- Se pretende, específicamente:
 - Determinar qué leyes sigue el software cuando evoluciona
 - Determinar qué leyes debe seguir el software para sobrevivir



Tipos de Sistemas de Lehman (I)

- Lehman (1980) distingue entre tres tipos de sistemas software
 - Atendiendo a la relación entre su especificación y su implementación
 - Especificación se entiende aquí en sentido genérico: desde una especificación formal hasta una simple descripción
- Sistemas Tipo-S (*specified*)
 - Aquellos que se han implementado siguiendo una especificación exacta de lo que el software debe hacer
 - Proporcionan una solución computacional para un problema dado, en un dominio abstracto y cerrado (p.e. matemático)
 - Por ejemplo, una librería o paquete que implemente de manera exacta un estándar definido de manera precisa
 - Ejemplo clásico: software que implementa la notación IEEE de punto flotante



Tipos de Sistemas de Lehman (II)

- Sistemas Tipo-P (*procedural*)
 - Aquellos que implementan un conjunto de procedimientos que se ejecutan de manera independiente y, en conjunto, desarrollan un comportamiento complejo
 - Cada uno de estos procedimientos está determinado, pero el comportamiento del conjunto no está especificado de manera exacta, sino que viene determinado por la combinación de los comportamientos de los elementos individuales
 - Ejemplo clásico: un software que juega al ajedrez
 - No varían sustancialmente con respecto al Tipo-S
- Sistemas Tipo-E (*evolving*)
 - Aquellos que pretenden realizar actividades relacionadas con el mundo real; es decir, que satisfacen una necesidad real



Sistemas Tipo-E

- Sistemas Tipo-E (cont.)
 - Al estar basados en el mundo real:
 - Su comportamiento refleja el del mundo real
 - Este comportamiento está *fuertemente ligado* al entorno en el que el software se ejecuta
 - Por tanto, el software debe adaptarse a los cambios que surjan en ese entorno, y que toman la forma de:
 - Variaciones de los requisitos
 - Nuevas circunstancias del entorno
 - En resumen: el software está ligado a la realidad, y debe cambiar a medida que esta realidad cambia
 - Ejemplo clásico: sistemas empotrados (cambio de hw)
 - En realidad, la mayoría del software está en este grupo



Leyes de Lehman (I)

- Las leyes se aplican solamente a los sistemas tipo-E
 - Es decir, la mayoría del software (el que es relevante)
- Primera Ley de Lehman (1974)

Ley del Cambio Continuo

- Todo sistema tipo-E que se utilice debe ser *adaptado de manera continua* o, de lo contrario, se irá haciendo progresivamente menos satisfactorio.
-
- Esto se debe a dos motivos:
 - El mundo real también cambia, de manera inevitable
 - El software nunca describe el dominio operacional deseado de manera exacta (“Principio de Incertidumbre del Software”)



Leyes de Lehman (2)

- Segunda Ley de Lehman (1974)

Ley de la Complejidad Creciente (o Incremental)

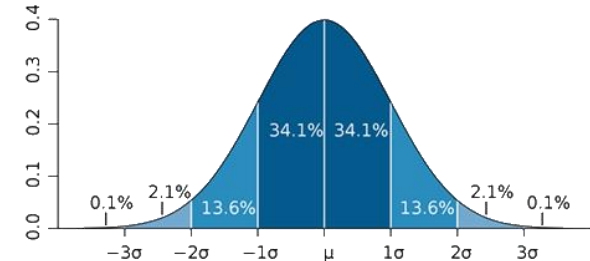
- A medida que un sistema tipo-E evoluciona, su complejidad se incrementa, a menos que específicamente se trabaje para mantenerla, o incluso para reducirla
- A medida que el entorno operacional cambia:
 - Cada cambio supone un nuevo tipo de interacción
 - No sólo hay más interacciones (más complejidad), sino que muchas son *añadidos* sobre el diseño inicial
 - El esfuerzo para combatir este efecto (reingeniería) supone menos capacidad para crear nuevas funcionalidades
- Es similar a la Segunda Ley de la Termodinámica ☺



Leyes de Lehman (3)

- Tercera Ley de Lehman (1974)

Ley de la Auto-Regulación



- Los procesos de evolución de los sistemas tipo-E son autorregulados, de modo que las métricas de los atributos de proceso y producto son cercanas a la distribución normal
- Los gestores se aseguran de que se cumplen los objetivos de la organización y las restricciones operacionales, introduciendo equilibrios y realizando comprobaciones
- Esto causa un crecimiento controlado por mecanismos de realimentación y estabilización
- Los parámetros de la dinámica resultante siguen (al menos en parte) una distribución normal



Leyes de Lehman (4)

- Cuarta Ley de Lehman (1978)

Ley de la Conservación de la Estabilidad Organizacional

- o del *Ritmo de Trabajo Invariante*

- La media efectiva del ritmo global de actividad (esfuerzo total empleado) en un sistema tipo-E que evoluciona es *invariante* a lo largo del tiempo de vida del producto

- A primera vista, esto no parece tener ningún sentido
- Sin embargo, todo intento por incrementar la productividad es contrarrestado por fuerzas de signo contrario
- La Ley de Brooks se muestra como un caso particular
 - (Aumentar el número de miembros de un equipo aumenta el *overhead* de comunicación entre ellos)



Leyes de Lehman (5)

- Quinta Ley de Lehman (1978)
 - **Ley de la Conservación de la Familiaridad**
 - Durante la vida activa de un sistema tipo-E que evoluciona, el ritmo medio de crecimiento (*incremental growth*) permanece invariante: es decir, el incremento en contenido entre las sucesivas versiones es estadísticamente invariante
 - Uno de los factores que determina el éxito de un sistema es el grado de familiaridad que todos los actores implicados (esto es, que comparten sus objetivos) tienen con él
 - Todos deben alcanzar cierto nivel de competencia con él
 - Cuantos más cambios haya en cada versión, más difícil será que todos sean conscientes de ellos, los entiendan y los aprecien
 - Más difícil es mantener ese nivel de competencia



Leyes de Lehman (6)

- Sexta Ley de Lehman (1991)

Ley del Crecimiento Continuo

- El contenido funcional (la funcionalidad) de un sistema tipo-E debe *incrementarse de manera continua* durante todo su tiempo de vida para *mantener* el grado de satisfacción de los usuarios
- A pesar de la similitud aparente, no confundir con la Primera Ley
 - Aquélla derivaba de una falta de correspondencia entre el mundo real (cambiante) y el software (que debe cambiar)
- Esta indica un doble efecto (que se realimenta)
 - Los usuarios demandan más funcionalidad
 - Es decir, el contenido funcional se incrementa
 - Con el tiempo, se demanda *cada vez* más funcionalidad
 - Por tanto, el contenido funcional se incrementa cada vez más



Leyes de Lehman (7)

- Séptima Ley de Lehman (1997)

Ley de la Calidad Decreciente

- En un sistema de tipo-E, se percibe que la calidad decae (se reduce) con el tiempo, a menos que se mantenga de manera rigurosa, y que se adapte a los cambios del entorno operacional
- No se trata de que *realmente* tenga menos calidad, sino que se *percibe* de este modo
 - La pérdida de calidad es aparente, en primera instancia
- Sin embargo, tiene efectos totalmente reales
 - El sistema no se adapta a cambios, y “se queda” viejo
 - Se percibe como viejo, y menos útil: y termina siéndolo



Leyes de Lehman (8)

- Octava Ley de Lehman (1997)

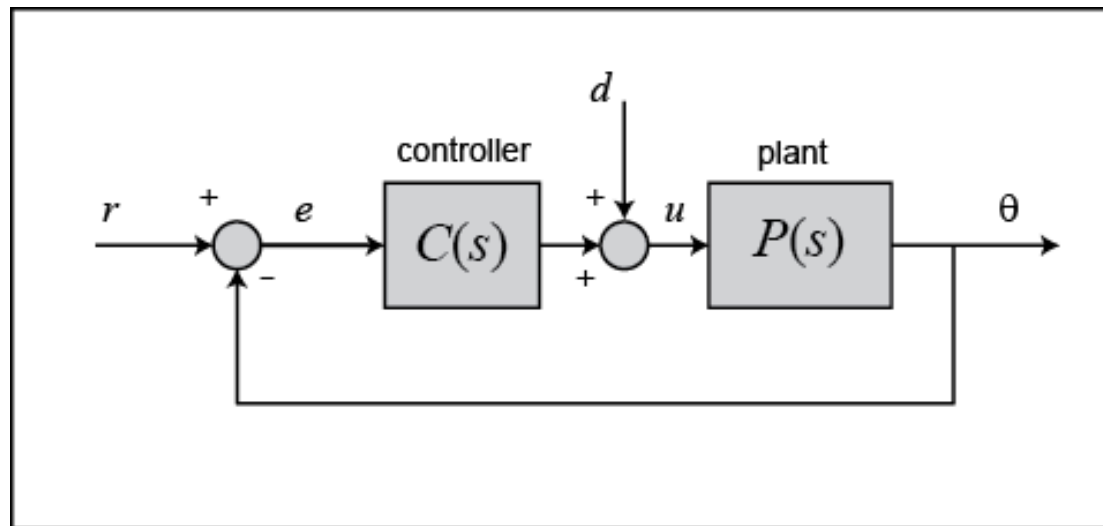
Ley del Sistema de Retroalimentación (Feedback)

- Los procesos de evolución de los sistemas de tipo-E se constituyen como sistemas de retroalimentación multi-nivel, con múltiples agentes y múltiples lazos (*loops*); y deben tratarse como tales si se desea lograr en ellos cualquier mejora significativa sobre cualquier base razonable
- Multi-nivel significa que la realimentación se da en más de un aspecto del software (e incluso de su documentación)
- La referencia a “lazos múltiples” (*multi-loop*) indica que se trata de un proceso iterativo, en el que se alcanza el equilibrio después de un cierto número de reajustes (iteraciones)



Octava Ley de Lehman

- Sistema de Retroalimentación

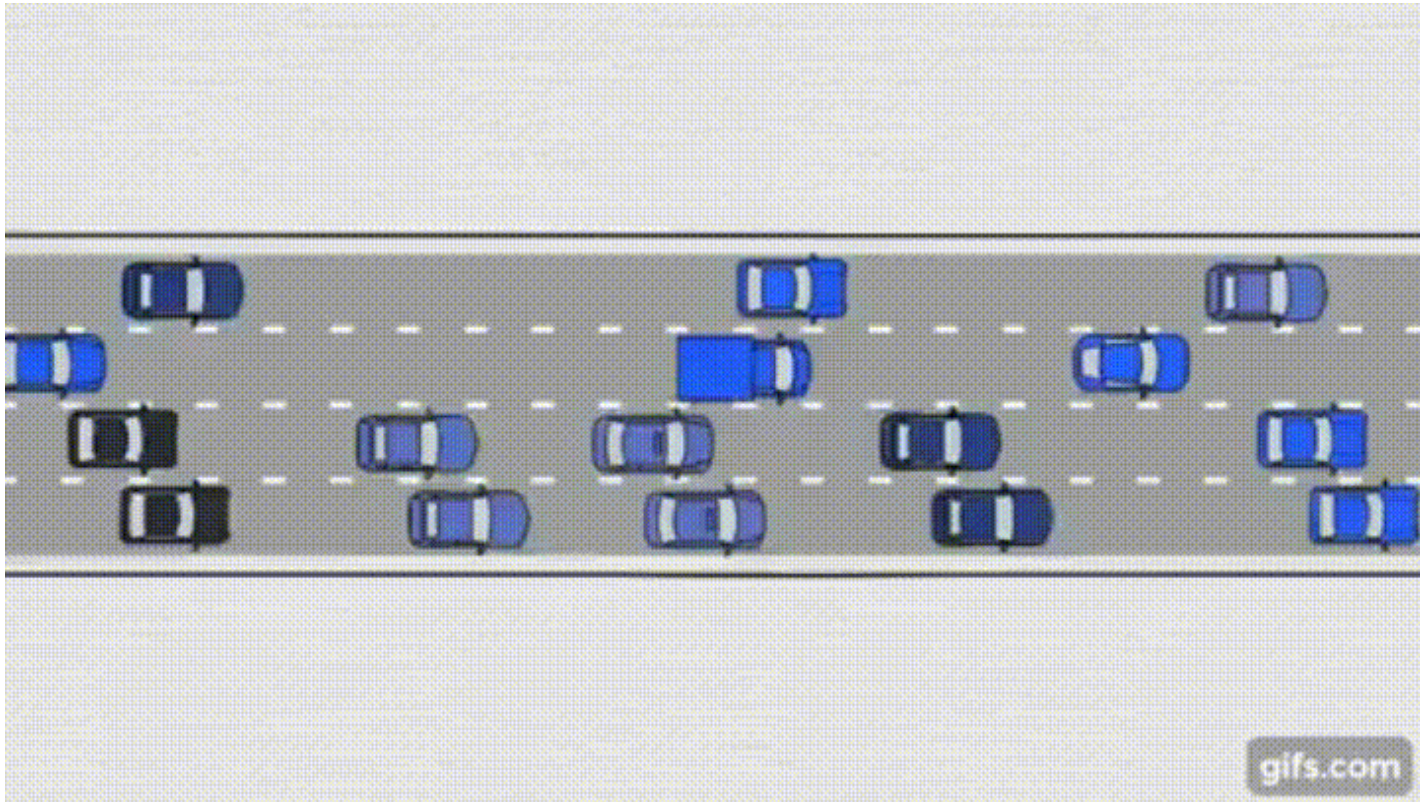


- El esquema multi-nivel, multi-agente y multi-lazo es mucho más complejo, como es lógico



Sistema de Feedback

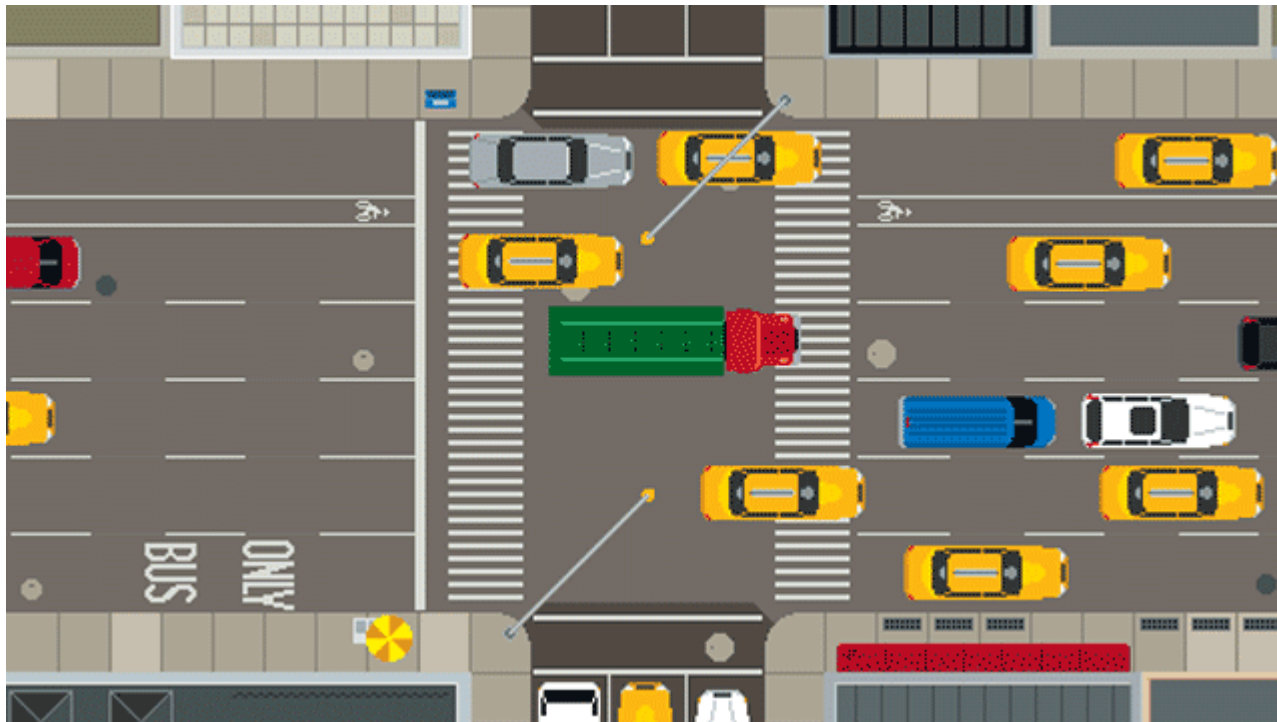
- Los sistemas se retroalimentan de manera natural
 - Con efectos positivos o negativos





Sistema de Control (Lazo Abierto)

- Pero incluso un sistema abierto puede ser controlado
 - Los semáforos son un mecanismo de control sencillo
 - Pero tremendamente eficaz





Sistema de Control Retroalimentado

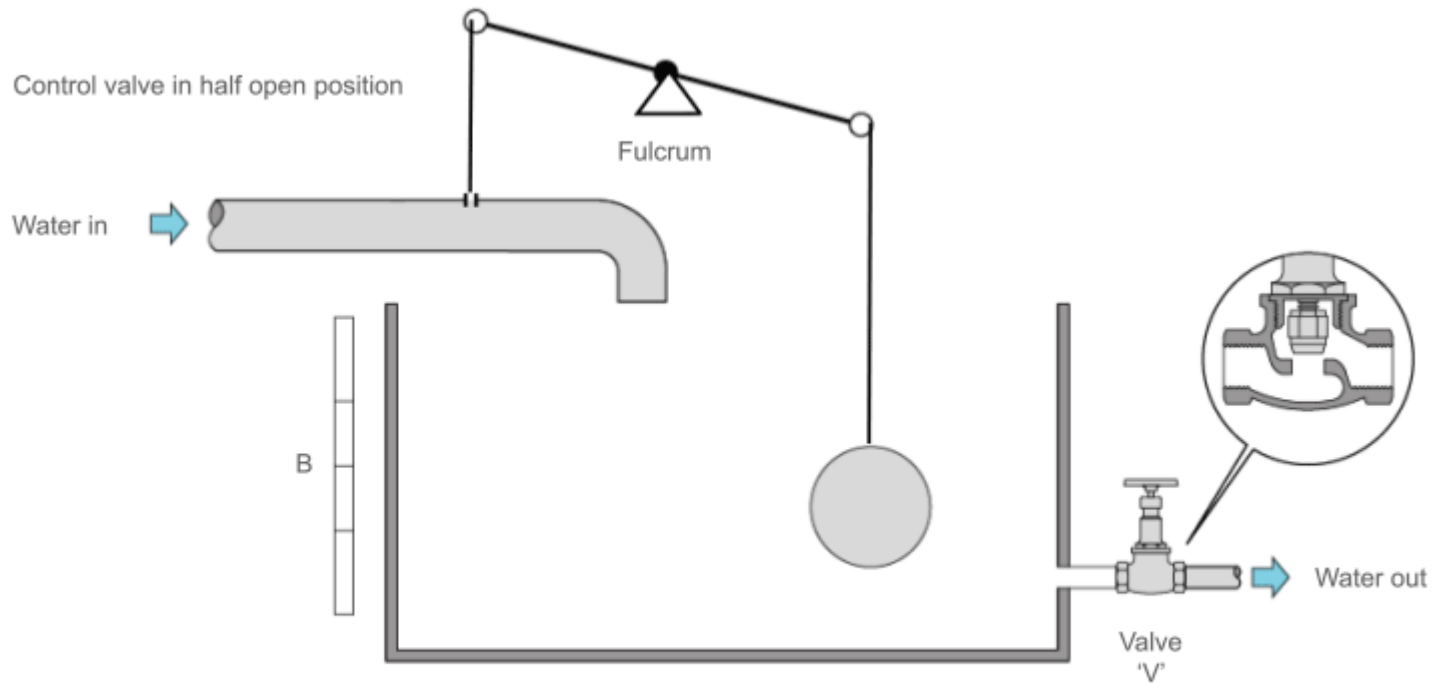


Fig. 5.2.4 Valve 50% open

- Sistema en equilibrio (flujo de E/ = flujo de /S)



Leyes de Lehman

- No son “leyes” en el sentido normativo.
 - Nadie obliga a que las cosas sean de esta manera
 - No son inevitables: se pueden corregir mediante educación, una vez identificados los efectos negativos
- Se denominan “leyes” en un sentido similar al que se da en ciencias
- Se deducen por observación del mundo real
 - Originalmente, Lehman los dedujo del desarrollo de la evolución del legendario sistema operativo OS/360 de IBM
 - Pero no derivan sólo de cuestiones técnicas o relativas al software, por lo que trascienden el ámbito de la ingeniería de software (es decir, escapan a su control)
- Por tanto, desde nuestro punto de vista son “Leyes” que se aplican globalmente en la ingeniería de software



Universidad
Rey Juan Carlos

