

BIBLIOTECA VIRTUAL

Parte V

# Conclusiones



## Capítulo 9

# Conclusiones y Trabajo Futuro

### 9.1 Conclusiones

La premisa básica de la Arquitectura de Software Dinámica es que los aspectos dinámicos y evolutivos de un sistema son parte integrante de su arquitectura, por lo que han de ser modelados junto a los aspectos estructurales, y a su mismo nivel. Sin embargo, la mayoría de los ADLs tradicionales han ignorado este aspecto, o lo han tratado de manera insuficiente. Incluso cuando se incluye, la especificación de dinamismo es casi siempre tratada como un aspecto secundario, y en muchos casos está considerablemente limitada.

Inicialmente, este trabajo de tesis comenzó como la búsqueda de un modelo general de dinamismo arquitectónico, que fuese capaz –al menos– de englobar a todas las propuestas ya existentes en un marco común. El punto clave de esta búsqueda era encontrar un concepto unificador, que permitiera considerar a los distintos enfoques parciales desde una perspectiva uniforme.

La hipótesis de este trabajo de tesis es que este concepto es el de Reflexión. Por tanto, se plantea como objetivo principal la elaboración de un marco reflexivo de Arquitectura de Software, que permita demostrar dicha hipótesis, esto es, que *la introducción de Reflexión en la Descripción Arquitectónica proporciona un marco general para la especificación de Dinamismo*. Creemos que este trabajo ha justificado esta hipótesis con amplitud suficiente.

En el proceso seguido para la consecución de este objetivo, se han obtenido los siguientes resultados intermedios:

- ☑ Se ha realizado un estudio general sobre el problema de la Arquitectura de Software Dinámica, haciendo especial énfasis en las distintas propuestas existentes para su solución. Cada una de estas propuestas ha sido examinada en detalle, y relacionada con otros enfoques similares según su origen conceptual. Como resultado de este estudio, se han identificado siete modelos básicos de dinamismo, que resumen todas las aproximaciones utilizadas, haciendo posible la comparación incluso entre las más dispares.
- ☑ Se han revisado las nociones de Reflexión desde la perspectiva de la Arquitectura de Software, basándose en los modelos de dinamismo identificados; de este modo, se han extraído

## 9. Conclusiones y Trabajo Futuro

---

aquellas que se consideran especialmente relevantes en el contexto, elaborando a partir de ellas un *modelo reflexivo de descripción arquitectónica* de carácter general.

- ☑ Con el fin de dar una descripción precisa de este modelo, se ha elaborado una definición formal completa del mismo, basada en teoría de conjuntos y de relaciones. Esta definición se ha realizado en tres etapas: en la primera ( $\aleph_{\Sigma}$ ) se han descrito todos los conceptos estándar de la Arquitectura de Software, incluyendo una formulación básica de tipos. En la segunda ( $\aleph_M$ ), se han introducido los conceptos reflexivos, incluyendo a la noción de *reificación* concebida como tercera dimensión de la descripción arquitectónica. Por fin, en la tercera ( $\aleph_{\Pi}$ ) se han colapsado parcialmente las relaciones de tipado y reificación, con el fin de facilitar la expresión de reflexión estructural. Al conjunto de estos (tres) modelos se le ha dado el nombre común de MARMOL. Incluso el primero de ellos constituye un trabajo original, pues es uno de los primeros estudios del campo desde esta perspectiva; los otros dos contribuyen a aclarar notablemente el impacto del concepto de reflexión desde un punto de vista estrictamente *estructural*.
- ☑ Con el fin de demostrar la aplicabilidad de este modelo, se ha construido un lenguaje de descripción con base en el mismo, al que se ha dado el nombre de *PiLar*. Aunque se concibe en primer lugar como un ADL reflexivo, se ha diseñado de forma que sea lo más general posible. El planteamiento general de su sintaxis lo sitúa como un ADL de tipo algebraico, con un conjunto de instrucciones minimal. Sólo dos instrucciones son específicamente dinámicas: su potencia se desarrolla mediante el esquema reflexivo.
- ☑ Se ha desarrollado una *semántica formal* para este lenguaje, proporcionando de este modo una descripción precisa de su significado. Esta semántica ha sido planteada en términos de una variante del cálculo- $\pi$  tipado, lo que demuestra la consistencia del enfoque utilizado. La reflexión ha sido planteada en términos de relaciones de reificación explícita entre estructuras –de manera similar al esquema descrito en MARMOL– y de relaciones de *superposición* entre procesos, planteado en términos del cálculo- $\pi$  como una fusión de ámbitos. El esquema global de la semántica es muy complejo, pero resulta totalmente coherente, tanto con la definición estructural del modelo, como con la teoría de tipos del cálculo.
- ☑ Se ha planteado, además, la posibilidad de incluir extensiones específicas en el lenguaje. Este aspecto se ha mostrado describiendo una de ellas, relacionada con la noción de *envolventes* reflexivos, que resulta especialmente adecuada en este contexto.
- ☑ Se ha comprobado la utilidad de *PiLar* como ADL de tipo general a través del desarrollo de una serie de ejemplos. De este punto se extraen las conclusiones siguientes:
  - Ante la descripción de sistemas estáticos, el uso de *PiLar* es tan sencillo como el de cualquier otro ADL existente. De hecho, sus características lingüísticas específicas lo hacen más potente que muchos de ellos.
  - En líneas generales, la descripción del dinamismo habitual en otros lenguajes –creación y destrucción de componentes y conexiones– resulta natural y bastante cómoda. Algunos esquemas, no obstante, muestran una solución más compleja, derivada del hecho de que parten de unas premisas diferentes.

- El lenguaje supone un marco especialmente adecuado para la especificación de esquemas reflexivos, no comparable al de ningún otro lenguaje. También extrae con naturalidad los patrones reflexivos implícitos en arquitecturas aparentemente convencionales. No obstante, puede alcanzar una gran complejidad, en especial cuando se manipulan explícitamente más de dos niveles.
  - La reificación de conexiones permite un nuevo enfoque de la noción de conector –el *conector de nivel meta*–, que permite conciliar las posturas antagónicas existentes ante este concepto. Supone una aproximación totalmente diferente al mismo, y manifiesta un gran potencial que aún ha de ser explotado.
- ☑ Se ha de concluir, en definitiva, que el uso de reflexión no sólo permite establecer un modelo unificado para la expresión de dinamismo, sino que tiene aún más posibilidades que las inicialmente planteadas. Existe toda una serie de relaciones, incluso de tipo estático, que se expresan con mayor facilidad en la presencia de dinamismo. La descripción del concepto de *envolventes* es sólo una de ellas.
- ☑ Como objetivo secundario, se ha mostrado también que la inclusión de un álgebra de procesos tipo-CCS en un marco reflexivo dota al sistema de unas capacidades análogas a las del propio cálculo- $\pi$ . Esto no es sorprendente, en realidad, ya que la potencia de este último deriva, entre otros aspectos, del modo en que difumina –controladamente– la distinción uso/mención.

En definitiva, los objetivos planteados han sido alcanzados. Se ha logrado construir, en efecto, un modelo reflexivo de descripción arquitectónica, y se ha mostrado su aplicabilidad al problema de la descripción de dinamismo. Más aún, el modelo obtenido es lo bastante genérico como para trascender estos objetivos, de modo que define un marco apto para la descripción de otras abstracciones interesantes, no necesariamente relacionadas con el dinamismo.

### 9.2 Líneas de Trabajo Futuro

El objetivo principal de este trabajo era llegar a la definición de un modelo arquitectónico general (MARMOL) y un lenguaje de descripción específico (*PiLar*), completamente formalizado, que demostrasen la viabilidad de la aproximación reflexiva a la especificación de dinamismo. En este sentido, ambos aspectos han sido completamente desarrollados, y la potencia expresiva del enfoque reflexivo suficientemente justificada.

Sin embargo, una vez se ha elaborado este marco, resultará lógico utilizarlo para explorar otras abstracciones cercanas, así como algunos aspectos suplementarios de las aquí examinadas. De este modo, no sólo se alcanzará una mayor comprensión de las mismas, sino también de las posibilidades del entorno. En principio, parece que la expresividad del modelo reflexivo –y concurrente– lo hace apto para la descripción de casi cualquier estructura. Algunas de ellas son particularmente interesantes en sí mismas, y su estudio podría determinar tanto los límites como la adecuación del enfoque utilizado.

Por otra parte, hay aspectos, tanto de la descripción arquitectónica como del soporte formal empleado, que podrían haberse expandido en más de un sentido, llevando a algunos conceptos

## 9. Conclusiones y Trabajo Futuro

---

hasta sus últimas consecuencias. Recorrer algunos de estos caminos permitiría ampliar el dominio de aplicación del modelo y el lenguaje definidos, así como adquirir una mayor comprensión de los aspectos implicados.

Estas consideraciones dividen a las líneas de trabajo futuro en dos grupos. En el primero, relativo al estudio de abstracciones en el marco, se considerarán las siguientes:

- Una abstracción para la que este modelo define un entorno particularmente adecuado es la de *interacción múltiple* en entornos concurrentes [FF96, JS96]. En realidad, podría decirse que, en este sentido, el concepto y el modelo siguen esquemas análogos. La interacción concurrente se plantea como primitiva, y se especifica después en términos de interacciones binarias. Del mismo modo, en *PiLar* se puede definir una conexión múltiple, y sólo se desarrolla su estructura –de nuevo, como interacciones binarias– en el nivel meta.
- Además de esto, se plantea el interés de realizar un estudio general de la noción de *intermediario*, aprovechando el marco especialmente adecuado que proporciona el concepto de conector de nivel meta. Esto incluiría desde los conectores clásicos de la Arquitectura de Software, hasta los *adaptadores* de Canal [CPT99], pasando por los *sincronizadores* y *meta-actores* de Frølund [FA93], los *envolventes* [Höl93] en cualquiera de sus encarnaciones, o en general todo tipo de *mediadores* [DeL99, GHJ+95a, Sch99, Sul94]. Un estudio de este tipo permitiría comprobar las analogías y diferencias entre todos ellos, y arrojaría nueva luz sobre el propio concepto de *conector*.
- Aún más interesante resulta la exploración de la analogía entre los componentes reflexivos de *PiLar* y la noción de *prototipos* [Bor86, Car96, DMC92, Foo89, US87, Ung88], en particular en lo que respecta al dinamismo de tercer nivel. Es habitual utilizar implícitamente una analogía entre componentes y objetos, que asimila a los arquetipos con las clases; sin embargo, en un enfoque dinámico como éste, en el que una instancia puede alterar su comportamiento e incluso “cambiar” de tipo por reificación, la analogía con el modelo de prototipos es aún más natural. No obstante, esta analogía no es total, dado que en *PiLar* las instancias siguen manteniendo un vínculo con su patrón estructural –su “clase”– a través de la reificación. En cualquier caso, un estudio de este tipo contribuiría a clarificar la naturaleza de las relaciones entre los elementos de *PiLar*.
- En esta misma línea, uno de los aspectos más interesantes a explorar se encuentra en la definición semántica de *PiLar*. En su forma actual, una reificación afecta a la estructura y proceso propios de cada componente. Si éste es un metacomponente, el efecto de la reificación se extiende a sus avatares. Sin embargo, las alteraciones no se propagan a su brhátar, esto es, a la definición pasiva del arquetipo correspondiente. Esto significa que los cambios afectan a las instancias existentes, pero no a las que se creen en el futuro. Se podría plantear el enfoque opuesto –de hecho, su especificación semántica sería notablemente sencilla–, pero en tal caso no podría recuperarse la definición de tipo original. Ambos enfoques tienen sus ventajas e inconvenientes, por lo que su combinación podría ser ventajosa: no obstante, esto implicaría tener dos formas de expandir la reificación, lo que sería sin duda complicado.

- Finalmente, un detalle que ya ha sido comentado es la particular adecuación de la estructura de reificaciones múltiples de  $\mathcal{P}i\mathcal{L}ar$  a los conceptos del denominado *desarrollo de software orientado a aspectos* (AOSD) [OK02]. Es obvio que se puede plantear la definición de aspectos –sujetos, dimensiones– como componentes de nivel meta, y luego reificarlos simultáneamente sobre un mismo avatar, consiguiendo de este modo una combinación –tejido– de aspectos a partir de la propia dinámica concurrente de las reglas de  $\mathcal{P}i\mathcal{L}ar$ . Éste es, de hecho, el mismo enfoque de James Andrews [And01] para la especificación de aspectos en términos de un álgebra de procesos. Por otra parte, en términos más puramente arquitectónicos, se puede considerar que un esquema de este tipo constituiría la primera definición explícita de *vistas* (§2.4.6) en un ADL.

En cuanto a las posibles extensiones del trabajo realizado, que constituyen el segundo grupo, se considerarán las siguientes:

- Un concepto arquitectónico que ha sido explícitamente ignorado es el de estilo. Una de las primeras extensiones a plantear es, por tanto, la *inclusión de estilos*. Se ha dicho (§2.4.5) que en realidad este aspecto no ha sido satisfactoriamente acometido por ningún ADL, aunque sin duda el enfoque de Wright [All97] es uno de los más interesantes. Allí, las restricciones sobre el concepto de estilo se expresaban en lógica de predicados. Ese mismo enfoque podría ser aplicado en  $\mathcal{P}i\mathcal{L}ar$ ; no obstante, en un contexto dinámico y concurrente, parecería más apropiado utilizar alguna forma de lógica modal, en especial una lógica temporal, ya que permite especificar con mayor facilidad las propiedades deseadas [ADG98]. La adecuación de éstas con un sistema de álgebra de procesos, como el utilizado en  $\mathcal{P}i\mathcal{L}ar$ , resulta especialmente apropiada [BS01], como demuestra el uso de una lógica modal específica en el MWB [VM94], o el del cálculo- $\mu$  modal en el CWB.
- En esta tesis, se define un componente en base al concepto de *bisimulación débil*. Esto significa, simplemente, que su comportamiento se define en base a las acciones que pueden observarse desde el exterior. Por tanto, toda entidad cuyo comportamiento observable respete –equivalga a– las restricciones definidas para un componente dado podrá ser utilizada en su contexto. No obstante, esta definición puede considerarse demasiado restrictiva, en el sentido de que no plantea la posibilidad de la extensión o sustitución de los componentes. En este sentido, resultaría interesante explorar una noción de subtipado para componentes. Un desarrollo inicial de este aspecto podría basarse en los estudios preliminares de Dinesh Katiyar [KLM94] y David Rosenblum [MRT98], y muy especialmente en la noción de *compatibilidad* descrita por Carlos Canal [CPT01] en el contexto de LEDA.
- En esta tesis, se ha discutido informalmente el modo en que la reflexión y la *superposición* se corresponden en un modelo concurrente, y se han citado otras opiniones similares [For00, KG99] en esta línea. Sin embargo, no se ha desarrollado todavía un estudio detallado en este sentido, que resultaría particularmente interesante. El modelo definido en esta tesis se muestra como un entorno particularmente adecuado para realizar tal comparación:  $\mathcal{P}i\mathcal{L}ar$  ha sido descrito como un ADL reflexivo. No obstante, puede demostrarse fácilmente que además permite expresar una noción de superposición [Bos99, SK02] entre componentes, sin necesidad de cambio alguno en su estructura.

## 9. Conclusiones y Trabajo Futuro

---

- Desde un punto de vista principalmente teórico, y una vez que se ha dado una noción de componente basada en la equivalencia observacional, resulta interesante plantear un estudio del modelo desde la perspectiva opuesta, utilizando un enfoque coalgebraico [JR97, RT94]. Esto supone, por una parte, un desarrollo completamente distinto; sin embargo, ambos muestran puntos de coincidencia notables, a través de la noción de bisimulación. Por otra parte, la perspectiva que se obtendría de la reflexión desde un punto de vista coinductivo es un aspecto particularmente intrigante. Algunos trabajos ya existentes [Bar00, Jac98] podrían ser utilizados como base inicial para un estudio de este tipo.
- De manera similar, resulta especialmente interesante el acometer un estudio de MARMOL basado en teoría de categorías. Es sencillo pasar desde la descripción actual a una en términos de teoría de grafos; ésta podrá luego expandirse teniendo en cuenta las nociones categóricas. Algunos aspectos en particular, como la correspondencia entre instancias y tipos de componente o el colapso –metalepsis– de las nociones de reificación y tipado, se beneficiarían enormemente de esta perspectiva. En este sentido, resulta también interesante la comparación con el enfoque de Michel Wermelinger [Wer99], que constituye el trabajo más completo en esta línea.
- Finalmente, se ha de considerar que el esquema de reificaciones, tal como se establece en *PiCar*, muestra una estructura parcialmente dispersa. Las reificaciones se establecen libremente entre componentes individuales, y no se adquiere una visión global de su estructura, salvo deduciéndola a partir de la especificación. Como señala Medvidovic [MT00], precisamente uno de los rasgos más significativos de la Arquitectura de Software es hacer patente los esquemas de interacción mediante una noción explícita de configuración. Del mismo modo, sería deseable describir un esquema similar para la reificación, definiendo una noción análoga de *meta-configuración*. Sin embargo, aún no está claro como acometer un planteamiento de este tipo, ya que cualquier construcción en este sentido va explícitamente en contra de la flexibilidad del modelo, que es su valor principal.

Las líneas de trabajo indicadas sólo son algunas de las posibles; por supuesto, podrían plantearse muchas otras. Por ejemplo, un enfoque interesante sería estudiar la sustitución del cálculo- $\pi$  por otro cálculo nominal: algunas de las características del cálculo de ambientes, en particular, parecen adaptarse especialmente al contexto de la descripción arquitectónica. No obstante, realizar –duplicar– este trabajo supondría una extensión considerable, y se trataría, en realidad, de una aproximación conceptualmente distinta.

# Bibliografía

- [ACDG95] Massimo Ancona, Walter Cazzola, Gabriella Dodero y Vittoria Gianuzzi. Channel Reification: a Reflective Approach to Fault Tolerant Software Development. En *Proceedings of OOPSLA '95*, pág. 137. ACM, Octubre 1995.
- [ACH98] Farhad Arbab, Paolo Ciancarini y Chris Hankin. Coordination Languages for Parallel Programming. *Parallel Computing*, 24(7):989–1004, 1998. Special Issue on Coordination.
- [ACPP91] Martín Abadi, Luca Cardelli, Benjamin Pierce y Gordon Plotkin. Dynamic Typing in a Statically Typed Language. *ACM Transactions on Programming Languages and Systems*, 13(2):237–268, Abril 1991.
- [ADG97] Robert Allen, Rémi Douence y David Garlan. Specifying Dynamism in Software Architectures. En *Proceedings of Foundations of Component-Based Systems Workshop*, Septiembre 1997.
- [ADG98] Robert Allen, Rémi Douence y David Garlan. Specifying and Analyzing Dynamic Software Architectures. En *Proceedings of 1998 Conference on Fundamental Approaches to Software Engineering*, volumen 1382 de *Lecture Notes in Computer Science*, págs. 21–37, Lisboa, Marzo 1998. Springer Verlag.
- [AF01] Luis F. Andrade y José Luiz Fiadeiro. Coordination Technologies for Managing Information Systems Evolution. En K. Dittrich, A. Geppert y M. Norrie, editores, *Proceedings of CAiSE'01*, volumen 2068 de *Lecture Notes in Computer Science*, págs. 374–387. Springer Verlag, 2001.
- [AG94a] Robert Allen y David Garlan. Beyond Definition/Use: Architectural Interconnection. *ACM SIGPLAN Notices*, 29(8), Agosto 1994. Proceedings of ACM IDL Workshop.
- [AG94b] Robert Allen y David Garlan. Formal Connectors. Technical Report CMU-CS-94-115, Carnegie Mellon University, 1994.
- [AG94c] Robert Allen y David Garlan. Formalizing Architectural Connection. En *Proceedings of the ICSE'16*, Mayo 1994.
- [AG99] Martín Abadi y Andrew D. Gordon. A Calculus for Cryptographic Protocols: The Spi Calculus. *Journal of Information and Computation*, 148(1):1–70, 1999.

- [Agh97] Gul A. Agha. Abstracting Interaction Patterns: A Programming Paradigm for Open Distributed Systems. En E. Najm y J.B. Stefani, editores, *Formal Methods for Open Object-based Distributed Systems*, IFIP Transactions. Chapman & Hall, 1997.
- [AHR93] Farhad Arbab, Ivan Herman y Eric P. M. B. Rutten. The Skeleton of a Computer Farm in Manifold. En H. El-Rewini, T. Lewis y B.D. Schriver, editores, *Proceedings of the 26<sup>th</sup> Annual Hawaii International Conference on System Sciences (HICSS26)*, volumen II, págs. 347–356, Los Alamitos, California, 1993. IEEE CS Press.
- [AHS93] Farhad Arbab, Ivan Herman y Per Spilling. An Overview of Manifold and its Implementation. *Concurrency: Practice and Experience*, 5(1):23–70, Febrero 1993. Also Report CS-R9142.
- [AIS<sup>+</sup>80] Christopher Alexander, Sarah Ishikawa, Murray Silverstein, Max Jacobson, Ingrid Fiksdahl-King y Shlomo Angel. *Un Language de Patrones*. Gustavo Gili, 1980.
- [AJ94] Samson Abramsky y Achim Jung. Domain Theory. En Samson Abramsky, Dov M. Gabbay y Tom S. E. Maibaum, editores, *Handbook of Logic in Computer Science*, volumen 3, págs. 1–168. Oxford University Press, 1994.
- [AKHS00] Colin Atkinson, Thomas Kühne y Brian Henderson-Sellers. To Meta or Not to Meta— That Is the Question. *Journal of Object-Oriented Programming*, págs. 32–35, Diciembre 2000.
- [AKN00] Franz Achemann, Stefan Kneubuehl y Oscar M. Nierstrasz. Scripting Coordination Styles. En Porto y Roman [PR00].
- [Ale81] Christopher Alexander. *El Modo Intemporal de Construir*. Gustavo Gili, 1981.
- [All97] Robert Allen. *A Formal Approach to Software Architecture*. Tesis Doctoral, School of Computer Science, Carnegie Mellon University, Mayo 1997. Technical Report CMU-CS-97-144.
- [ALP99] Uwe Aßmann, Andreas Ludwig y Daniel Pfeifer. Programming Connectors in an Open Language. En Clements et al. [CPR99].
- [ALSN00] Franz Achemann, Markus Lumpe, Jean-Guy Schneider y Oscar M. Nierstrasz. Piccola – A Small Composition Language. En Howard Bowman y John Derrick, editores, *Formal Methods for Distributed Processing: an Object-Oriented Approach*. Cambridge University Press, 2000.
- [AN00] Franz Achemann y Oscar M. Nierstrasz. Applications = Components + Scripts – A Tour of Piccola. En Mehmet Aksit, editor, *Software Architecture and Component Technology*, Enschede, Netherlands, 2000. Kluwer Academic Publishers.

- 
- [And91] Gregory Andrews. *Concurrent Programming. Principles and Practice*. Benjamin-Cummings, 1991.
- [And01] James H. Andrews. Process-Algebraic Foundations of Aspect-Oriented Programming. En Yonezawa y Matsuoka [YM01].
- [Arb96] Farhad Arbab. The IWIM Model for Coordination of Concurrent Activities. En Ciancarini y Hankin [CH96], págs. 24–56.
- [Arb98] Farhad Arbab. What Do You Mean, Coordination? *Bulletin of Dutch Association for Theoretical Computer Science (NVTI)*, Marzo 1998.
- [ASA01] Mark Astley, Daniel C. Sturman y Gul A. Agha. Customizable Middleware for Modular Distributed Software. *Communications of the ACM*, 44(5):99–107, Mayo 2001.
- [Ast99] Mark Christopher Astley. *Customization and Composition of Distributed Objects: Policy Management in Distributed Software Architectures*. Tesis Doctoral, University of Illinois at Urbana-Champaign, 1999.
- [AT02] Farhad Arbab y Carolyn Talcott, editores. *Fifth International Conference on Coordination Models and Languages, COORDINATION 2002*, volumen 2315 de *Lecture Notes in Computer Science*, York, Abril 2002. Springer Verlag.
- [AWB<sup>+</sup>93] Mehmet Aksit, Ken Wakita, Jan Bosch, Lodewijk Bergmans y Akinori Yonezawa. Abstracting Object Interactions using Composition Filters. En Rachid Guerraoui, Oscar Nierstrasz y Michel Riveill, editores, *Object-Based Distributed Programming*, volumen 791 de *Lecture Notes in Computer Science*, págs. 152–184. Springer-Verlag, 1993. ECOOP'93.
- [AWG99] Recommended Practice for Architectural Description. IEEE Architecture Working Group, Diciembre 1999. Draft: Version 5.2.
- [BA00] Lodewijk M. J. Bergmans y Mehmet Aksit. Aspects & Crosscutting in Layered Middleware Systems. En *Reflective Middleware 2000*. ACM/IFIP, 2000.
- [Bal97] Bob Balzer. Instrumenting, Monitoring & Debugging Software Architectures. Technical report, USC/ISI, 1997.
- [Bar92] Henk Barendregt. Lambda Calculi with Types. En Samson Abramsky, Dov M. Gabbay y Tom S.E. Maibaum, editores, *Handbook of Logic in Computer Science*, volumen 2, págs. 117–309. Oxford University Press, 1992.
- [Bar00] Luis Manuel Soares Barbosa. Components as Processes: An Exercise in Coalgebraic Modeling. En Scott F. Smith y Carolyn L. Talcott, editores, *4<sup>th</sup> Formal Methods for Open Object-Based Distributed Systems (FMOODS'2000)*, págs. 397–417, Stanford, 2000. Kluwer Academic Publishers.

- [BB90] Gérard Berry y Gérard Boudol. The Chemical Abstract Machine. En *POPL '90. Proceedings of the 17<sup>th</sup> annual ACM symposium on Principles of Programming Languages*, págs. 81–94, New York, NY, USA, 1990. ACM Press.
- [BB98] Lynne Blair y Gordon S. Blair. The Impact of Aspect-Oriented Programming on Formal Methods. En *Proceedings of ECOOP'98 Aspect-Oriented Programming Workshop*. Springer-Verlag, 1998.
- [BBI<sup>+</sup>00] Gordon S. Blair, Lynne Blair, Valérie Issarny, Petr Tůma y Apostolos Zarras. The Role of Software Architecture in Constraining Adaptation in Component-based Middleware Platforms. En Springer-Verlag, editor, *Proceedings of Middleware 2000*, Lecture Notes in Computer Science. IFIP/ACM, Abril 2000.
- [BC98] Gordon S. Blair y Geoff Coulson. The Case for Reflective Middleware. Technical Report MPG-98-38, Distributed Multimedia Research Group, Department of Computing, Lancaster University, 1998.
- [BCA<sup>+</sup>01] Gordon S. Blair, Geoff Coulson, Anders Andersen, Lynne Blair, Michael Clarke, Fabio Costa, Hector Duran-Limon, Tom Fitzpatrick, Lee Johnston, Rui Moreira, Nikos Parlavantzas y Katia Saikoski. The Design and Implementation of Open ORB v2. *IEEE Distributed Systems Online*, 2(6), 2001.
- [BCKK01] Gordon Blair, Fabio Costa, Roy Campbell y Fabio Kon. Reflective Middleware. *IEEE Distributed Systems Online*, 2(5), 2001.
- [BCD01] Marco Bernardo, Paolo Ciancarini y Lorenzo Donatiello. Detecting Architectural Mismatches in Process Algebraic Descriptions of Software Systems. En Kazman et al. [KKV<sup>v</sup>V01], págs. 77–86.
- [BCK98] Len Bass, Paul Clements y Rick Kazman. *Software Architecture in Practice*. Addison-Wesley, 1998.
- [BCRP98] Gordon S. Blair, Geoff Coulson, Phillippe Robin y Michael Papatomas. An Architecture for Next Generation Middleware. En N. Davies, K. Raymond y J. Seitz, editores, *Proceedings of the IFIP International Conference on Distributed Systems Platforms and Open Distributed Processing (Middleware'98)*, Lecture Notes in Computer Science, Lake District, UK, 1998. Springer Verlag.
- [BDH<sup>+</sup>98] Manfred Broy, Anton Deimel, Jürgen Henn, Kai Koskimies, Frantisek Plasil, Gustav Pomberger, Wolfgang Pree, Michael Stal y Clemens Szyperski. What Characterizes a Software Component? *Software Concepts and Tools*, 19(1), 1998. Software Research Lab, Technical Report # J010.
- [BDWL91] Mario R. Barbacci, Dennis Doubleday, Charles B. Weinstock y Randall W. Lichota. Durra: An Integrated Approach to Software Specification, Modeling and Rapid Prototyping. Technical Report CMU/SEI-91-TR-21, Software Engineering Institute (SEI), Septiembre 1991.

- 
- [Ber94] Lodewijk Bergmans. *Composing Concurrent Objects: Applying Composition Filters for the Development and Reuse of Concurrent Object-Oriented Programs*. Tesis Doctoral, TRESE, University of Twente, Enschede, Netherlands, Junio 1994.
- [BF88] Luc Bougé y Nissim Francez. A Compositional Approach to Superimposition. En *15<sup>th</sup> Annual ACM Symposium on Principles of Programming Languages, POPL'88*, págs. 240–249, San Diego, 1988. ACM Press.
- [BF97] Manuel Barrio y Pablo de la Fuente. Software Architecture: Object vs. Process Approach. En *Proceedings of SCCC'97*. IEEE Computer Society Press, Noviembre 1997.
- [BF02] Marco Bernardo y Francesco Franzè. Endogenous and Exogenous Extensions of Architectural Types. En Arbab y Talcott [AT02], págs. 40–55.
- [BFP01] Jan A. Bergstra, Wan Fokink y Alban Ponse. Process Algebra with Recursive Operations. En Bergstra et al. [BPS01], capítulo 5, págs. 333–389.
- [BGZ97a] Nadia Busi, Roberto Gorrieri y Gianluigi Zavattaro. On the Turing Equivalence of Linda Coordination Primitives. *Electronic Notes in Theoretical Computer Science*, 7, 1997. Express'97.
- [BGZ97b] Nadia Busi, Roberto Gorrieri y Gianluigi Zavattaro. Three Semantics of the Output Operation for Generative Communication. En Garlan y Le Métayer [GLM97], págs. 205–219.
- [BGZ98] Nadia Busi, Roberto Gorrieri y Gianluigi Zavattaro. A Process Algebraic View of Linda Coordination Primitives. *Theoretical Computer Science*, 192(2):167–199, 1998.
- [BK96] J. A. Bergstra y P. Klint. The TOOLBUS Coordination Architecture. En Ciancarini y Hankin [CH96], págs. 75–88.
- [BL97a] Maria Lúcia Blanck Lisbôa. Arquiteturas de Meta-Nível. Simpósio Brasileiro de Engenharia de Software, Fortaleza, Octubre 1997.
- [BL97b] Maria Lúcia Blanck Lisbôa. Reflexão Computacional no Modelo de Objetos. II Simpósio Brasileiro de Linguagens de Programação, Campinas, Septiembre 1997.
- [BLM96] Jean-Paul Banâtre y Daniel Le Métayer. Gamma and the Chemical Reaction Model: Ten Years After. En *Coordination Programming: Mechanisms, Models and Semantics*, págs. 3–41. World Scientific Publishing, IC Press, 1996.
- [Boo96] Grady Booch. *Análisis y Diseño Orientado a Objetos con Aplicaciones*. Addison-Wesley/Díaz de Santos, 2 edición, 1996.

- [Bor86] Alan H. Borning. Classes Versus Prototypes in Object-Oriented Languages. En *Proceedings of the ACM/IEEE Fall Joint Computer Conference*, págs. 36–40, Dallas, TX, Noviembre 1986. IEEE.
- [Bos96a] Jan Bosch. Composition through Superimposition. En *ECOOP '96 Workshop on Composability Issues in Object-Oriented (CIOO '96)*, 1996.
- [Bos96b] Jan Bosch. Language Support for Component Communication in LayOM. En *ECOOP'96 Workshop on Component-Oriented Programming (WCOP)*, 1996.
- [Bos96c] Jan Bosch. Relations as Object Model Components. *Journal of Programming Languages*, 4:39–61, 1996.
- [Bos98] Jan Bosch. Specifying Frameworks and Design Patterns as Architectural Fragments. En Jian Chen, M. Li, Christine Mingis y Bertrand Meyer, editores, *27<sup>th</sup> Conference on Technology of Object-Oriented Languages and Systems (TOOLS Asia '98)*, Beijing, China, Septiembre 1998. IEEE Computer Society Press.
- [Bos99] Jan Bosch. Superimposition: A Component Adaptation Technique. *Information and Software Technology*, 41(5):257–273, Marzo 1999.
- [Bos00] Jan Bosch. *Design and Use of Software Architectures. Adopting and Evolving a Product-Line Approach*. Addison-Wesley, Mayo 2000.
- [Bou98] Gérard Boudol. The  $\pi$ -Calculus in Direct Style. *Higher-Order and Symbolic Computation*, 11:177–208, 1998. Previous version in the *Proceedings of POPL '97*, ACM Press, pages 228–241.
- [BPS01] Jan A. Bergstra, Alban Ponse y Scott A. Smolka, editores. *Handbook of Process Algebra*. Elsevier Science – North Holland, Amsterdam, 2001.
- [Bro96] Manfred Broy. Towards a Mathematical Model of a Component and Its Use. En *Proceedings of Componentware Users Conference 1996*. SIGS Publications, 1996.
- [Bru99] Roberto Bruni. *Tile Logic for Synchronized Rewriting of Concurrent Systems*. Tesis Doctoral, University of Pisa, 1999.
- [BS95] Manuel Barrio Solórzano. *Estudio de Aspectos Dinámicos en Sistemas Orientados al Objeto*. Tesis Doctoral, Universidad de Valladolid, 1995.
- [BS01] Julian C. Bradfield y Colin P. Stirling. Modal Logics and mu-Calculi: An Introduction. En Bergstra et al. [BPS01], págs. 293–330.
- [BW97] Martin Büchi y Wolfgang Weck. A Plea for Grey-Box Components. Technical Report TUCS-TR-122, Turku Centre for Computer Science, Finlandia, Septiembre 1997.
- [CA94] Christian Jürgen Callsen y Gul A. Agha. Open Heterogeneous Computing in ActorSpace. *Journal of Parallel and Distributed Computing*, 21:289–300, 1994.

- 
- [Cal94] Christian Jürgen Callsen. *Open Distributed Heterogeneous Computing*. Tesis Doctoral, Aalborg University, Marzo 1994.
- [Car96] Luca Cardelli. Object-based vs. Class-based Languages. Tutorial at the 9<sup>th</sup> ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI'96), Philadelphia, Pennsylvania, Abril 1996.
- [Car97] Luca Cardelli. Type Systems. En Allen B. Tucker, Jr., editor, *The Handbook of Computer Science and Engineering*, capítulo 103, págs. 2208–2236. CRC Press, 1997.
- [Car98] Antonio Carzaniga. *Architectures for an Event Notification Service Scalable to Wide-area Networks*. Tesis Doctoral, Politecnico di Milano, Diciembre 1998.
- [Cas01] Ilaria Castellani. Process Algebras with Localities. En Bergstra et al. [BPS01], capítulo 15, págs. 945–1045.
- [Caz98] Walter Cazzola. Evaluation of Object-Oriented Reflective Models. En *Proceedings of ECOOP98 Workshop on Reflective Object-Oriented Programming and Systems (EWROOPS'98)*, Julio 1998.
- [Caz00] Walter Cazzola. *Communication-Oriented Reflection: a Way to Open Up the RMI Mechanism*. Tesis Doctoral, Università degli Studi di Milano, Milano, Italia, Noviembre 2000.
- [CCL00a] Walter Cazzola, Shigeru Chiba y Thomas Ledoux, editores. *On-Line Proceedings of ECOOP'2000 Workshop on Reflection and Metalevel Architectures*, Nice, France, Junio 2000.
- [CCL00b] Walter Cazzola, Shigeru Chiba y Thomas Ledoux. Reflection and Meta-Level Architectures: State of the Art, and Future Trends. En *On-Line Proceedings of ECOOP'2000 Workshop on Reflection and Metalevel Architectures* [CCL00a].
- [CD97] Stephen Crane y Naranker Dulay. A Configurable Protocol Architecture for CORBA Environments. En *ISADS-97, Berlin*, 1997.
- [CD99] Juan Carlos Cruz y Stéphane Ducasse. A Group-Based Approach for Coordinating Active Objects. En Ciancarini y Wolf [CW99], págs. 355–370.
- [CDF<sup>+</sup>95] Stephen Crane, Naranker Dulay, Halldor Fosså, Jeff Kramer, Jeff Magee, Morris Sloman y Kevin Twidle. Configuration Management for Distributed Systems. En *Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management (ISINM 95)*, Mayo 1995.
- [CF67] Haskell Brooks Curry y Robert Feys. *Lógica Combinatoria*, volumen 24 de *Estructura y Función*. Tecnos, 1967. Traducido por Manuel Sacristán.

- [CFBS98] Carlos E. Cuesta, Pablo de la Fuente y Manuel Barrio-Solórzano. Una Perspectiva Arquitectónica de la Invocación Estática en CORBA. En Ambrosio Toval Álvarez y Joaquín Nicolás Ros, editores, *III Jornadas de Ingeniería del Software (JIS'98)*, págs. 219–230, Murcia, Noviembre 1998. Universidad de Murcia, Servicio de Publicaciones.
- [CFBS99] Carlos E. Cuesta, Pablo de la Fuente y Manuel Barrio-Solórzano. An Architectural Perspective of the Static Invocation in CORBA. En Clements et al. [CPR99].
- [CFBSB00] Carlos E. Cuesta, Pablo de la Fuente, Manuel Barrio-Solórzano y Encarnación Beato. Arquitectura de Software Dinámica Basada en Reflexión. En Carlos Delgado Kloos, Esperanza Marcos y José Manuel Marqués Corral, editores, *V Jornadas de Ingeniería del Software y Bases de Datos (JISBD'2000)*, págs. 203–214, Valladolid, Noviembre 2000. Universidad de Valladolid, Secretariado de Publicaciones.
- [CFBSB01] Carlos E. Cuesta, Pablo de la Fuente, Manuel Barrio-Solórzano y Encarnación Beato. Dynamic Coordination Architecture through the use of Reflection. En *Proceedings of 16<sup>th</sup> ACM Symposium on Applied Computing (SAC2001)*, ACM Proceedings, págs. 134–140, Las Vegas, NV, Marzo 2001. ACM Press.
- [CFBSB02a] Carlos E. Cuesta, Pablo de la Fuente, Manuel Barrio-Solórzano y Encarnación Beato. Coordination in a Reflective Architecture Description Language. En Arbab y Talcott [AT02], págs. 141–148.
- [CFBSB02b] Carlos E. Cuesta, Pablo de la Fuente, Manuel Barrio Solórzano y M. Encarnación Beato. Introducing Reflection in Architecture Description Languages. En Jan Bosch, Morven Gentleman, Christine Hofmeister y Juha Kuusela, editores, *Software Architecture: System Design, Development and Maintenance*, capítulo 9, págs. 143–156. Kluwer Academic Publishers, Montréal, Québec, Canada, Agosto 2002.
- [CFM98] Paolo Ciancarini, F. Franzè y Cecilia Mascolo. A Coordination Model to Specify Systems Including Mobile Agents. En *Proceedings of the Ninth International Workshop on Software Specification and Design*, págs. 96–105. IEEE Computer Society Press, 1998.
- [CG86] Nicholas Carriero y David Gelernter. How to Write Parallel Programs: a Guide to the Perplexed. *ACM Computing Surveys*, 21(3):323–357, 1986.
- [CG98] Luca Cardelli y Andrew D. Gordon. Mobile Ambients. En Maurice Nivat, editor, *Foundations of Software Science and Computational Structures*, volumen 1378 de *Lecture Notes in Computer Science*, págs. 140–155. Springer, 1998.
- [CG00] Luca Cardelli y Andrew D. Gordon. Mobile Ambients. *Theoretical Computer Science*, 240(1):177–213, 2000.

- 
- [CGG00] Luca Cardelli, Giorgio Ghelli y Andrew D. Gordon. Secrecy and Group Creation. En Catuscia Palamidessi, editor, *CONCUR 2000 - Proceedings of 11<sup>th</sup> International Conference on Concurrency Theory*, volumen 1877 de *Lecture Notes in Computer Science*, págs. 365–379, University Park, PA, USA, Agosto 2000. Springer.
- [CGZ95a] Nick Carriero, David Gelernter y Lenore Zuck. Bauhaus Linda. En Paolo Ciancarini, Oscar Nierstrasz y Akinori Yonezawa, editores, *Object-Based Models and Languages for Concurrent Systems*, LNCS 924, págs. 66–76. Springer-Verlag, 1995.
- [CGZ95b] Paolo Ciancarini, Roberto Gorrieri y Gianluigi Zavattaro. Generative Communication in Process Algebra. Technical Report UBLCS-95-7, Università di Bologna, Febrero 1995.
- [CGZ96] Paolo Ciancarini, Roberto Gorrieri y Gianluigi Zavattaro. Towards a Calculus for Generative Communication. En E. Najm y J. Stefani, editores, *Proceeding IFIP Conference on Formal Methods for Open Object-Based Distributed Systems*, págs. 289–306, Paris, France, 1996. IFIP Wg 6.1.
- [CH96] Paolo Ciancarini y Chris Hankin, editores. *First International Conference on Coordination Models, Languages and Applications (Coordination'96)*, volumen 1061 de *Lecture Notes in Computer Science*, Cesena, Italia, Abril 1996. Springer Verlag.
- [Che94] Shing-Chi Cheung. *Tractable and Compositional Techniques for Behaviour Analysis of Concurrent Systems*. Tesis Doctoral, Imperial College of Science, Technologie and Medicine. University of London, Febrero 1994.
- [Chi95] Shigeru Chiba. A Meta-Object Protocol for C++. *ACM SIGPLAN Notices*, 30(10):285–299, Octubre 1995. Proceedings of OOPSLA'95.
- [Chu40] Alonzo Church. A Formulation of the Simple Theory of Types. *Journal of Symbolic Logic*, 5:56–68, 1940.
- [Cia91] Paolo Ciancarini. PoliS: a Programming Model for Multiple Tuple Spaces. En Carlo Ghezzi y Gruia-Catalin Roman, editores, *Proceedings of 6<sup>th</sup> IEEE International Workshop on Software Specification and Design*, págs. 44–51, Como, Italia, Octubre 1991. IEEE Computer Society Press.
- [Cia93] Paolo Ciancarini. Distributed Programming with Logic Tuple Spaces. Technical Report UBLCS-93-7, Università di Bologna, 1993.
- [Cia97] Paolo Ciancarini. *Coordination Models, Languages, Architectures and Applications*. Università di Bologna, Octubre 1997.
- [CKL96] Shigeru Chiba, Gregor Kiczales y John Lamping. Avoiding Confusion in Metacircularity: The Meta-Helix. En *Proceedings of Second International Symposium*

- on Object Technologies for Advanced Software (ISOTAS'96)*, número 1049 en *Lecture Notes in Computer Science*, págs. 157–172. Springer Verlag, 1996.
- [Cle96] Paul C. Clements. A Survey of Architecture Description Languages. En *Proceedings of 8<sup>th</sup> International Workshop on Software Specification and Design*, Marzo 1996.
- [CM88] K. Mani Chandy y Jayadev Misra. *Parallel Program Design: A Foundation*. Addison-Wesley, 1988.
- [CM96a] Manuel G. Clavel y José Meseguer. Axiomatizing Reflective Logics and Languages. En *Proceedings of Reflection'96*, págs. 263–288, 1996.
- [CM96b] Manuel G. Clavel y José Meseguer. Reflection and Strategies in Rewriting Logic. En *Proceedings of First International Workshop on Rewriting Logic and its Applications*, número 4 en *Electronic Notes in Theoretical Computer Science*. Elsevier Science, Septiembre 1996.
- [CM98] Paolo Ciancarini y Cecilia Mascolo. Software Architecture and Mobility. En Jeff Magee y Dewayne E. Perry, editores, *Proceedings of the Third International Software Architecture Workshop*. ACM Press, Noviembre 1998.
- [CM99] Paolo Ciancarini y Cecilia Mascolo. Specification and Analysis of Component Based Software Architectures. En Clements et al. [CPR99].
- [CPR99] Paul C. Clements, Dewayne E. Perry y Alexander Ran, editores. *Electronic Proceedings of First Working IFIP Conference on Software Architecture (WICSA1)*. IFIP, Febrero 1999.
- [CPT99] Carlos Canal, Ernesto Pimentel y José María Troya. Specification and Refinement of Dynamic Software Architectures. En *Software Architecture*, págs. 107–126. Kluwer Academic Publishing, Febrero 1999.
- [CPT01] Carlos Canal, Ernesto Pimentel y José María Troya. Compatibility and Inheritance in Software Architectures. *Science of Computer Programming*, 41(2):105–138, 2001.
- [Cra97] John Stephen Crane. *Dynamic Binding in Distributed Systems*. Tesis Doctoral, Imperial College of Science, Technology and Medicine, University of London, Septiembre 1997.
- [CRC98] José Angel Carsí, Isidro Ramos y J. Hilario Canós. La Metaclase OASIS. En *Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes Software (IDEAS'98)*, págs. 181–192, Torres, Brasil, Abril 1998.
- [CSST98] Walter Cazzola, Andrea Savigni, Andrea Sosio y Francesco Tisato. Architectural Reflection: Bridging the Gap Between a Running System and its Architectural Specification. En *Proceedings of 6<sup>th</sup> Reengineering Forum (REF'98)*, págs. 12–1–12–6. IEEE, Marzo 1998.

- 
- [CSST99] Walter Cazzola, Andrea Savigni, Andrea Sosio y Francesco Tisato. Architectural Reflection: Concepts, Design, and Evaluation. Technical Report RI-DSI 234-99, DSI, Università degli studi di Milano, Mayo 1999.
- [CST00] Walter Cazzola, Robert J. Stroud y Francesco Tisato, editores. *Reflection and Software Engineering*, volumen 1826 de *Lecture Notes in Computer Science*. Springer-Verlag, Heidelberg, Alemania, Junio 2000.
- [CTKI00] Shigeru Chiba, Michiaki Tatsubori, Marc-Olivier Killijian y Kozo Itano. Open-Java: a Class-based Macro System for Java. En Cazzola et al. [CST00], págs. 119–135.
- [CV00] Carlos Canal Velasco. *Un Lenguaje para la Especificación y Validación de Arquitecturas de Software*. Tesis Doctoral, Universidad de Málaga, 2000.
- [CW99] Paolo Ciancarini y Alexander L. Wolf, editores. *Third International Conference on Coordination Models and Languages (Coordination'99)*, volumen 1594 de *Lecture Notes in Computer Science*. Springer-Verlag, Abril 1999.
- [Dam95] Laurent Dami. Functions, Records and Compatibility in the  $\lambda N$  Calculus. En Oscar Marius Nierstrasz y Dionysios C. Tsihrizis, editores, *Object-Oriented Software Composition*, The Object-Oriented Series, capítulo 6, págs. 153–174. Prentice-Hall, 1995.
- [DC95] Thomas R. Dean y James R. Cordy. A Syntactic Theory of Software Architecture. *IEEE Transactions on Software Engineering*, 21(4):302–313, Abril 1995.
- [DC01a] Jim Dowling y Vinny Cahill. The K-Component Architecture Meta-Model for Self-Adaptive Software. En Yonezawa y Matsuoka [YM01], págs. 81–88.
- [DC01b] Jim Dowling y Vinny Cahill. The K-Component Model and Dynamic Software Evolution. En Siobhán Clarke, Simon Dobson, Vinny Cahill y Robert Filman, editores, *Proceedings of the OOPSLA 2001 workshop on Engineering Complex Object-Oriented Systems for Evolution (ECOOSE'01)*, Tampa Bay, Florida, Octubre 2001.
- [DCS76] Maria Luisa Dalla Chiara Scabia. *Lógica*. Labor, 1976.
- [DeL99] Robert DeLine. *Resolving Packaging Mismatch*. Tesis Doctoral, Carnegie Mellon University, Julio 1999.
- [DHN00] Stéphane Ducasse, T. Hofmann y Oscar Nierstrasz. OpenSpaces: an Object-Oriented Framework for Reconfigurable Coordination Spaces. En Porto y Catalin-Roman [PR00], págs. 1–19.
- [Dij68] Edsger Wybe Dijkstra. The Structure of the THE Multiprogramming System. *Communications of the ACM*, 11(5):341–346, Mayo 1968.

- [Dij75] Edsger W. Dijkstra. Guarded Commands, Nondeterminacy and Formal Derivation of Programs. *Communications of the ACM*, 18(8):453–457, Agosto 1975.
- [DM95] François-Nicola Demers y Jacques Malenfant. Reflection in Logic, Functional and Object-Oriented Programming: a Short Comparative Study. En *Proceedings of the IJCAI'95 workshop on Reflection and Meta-Level Architectures and their Applications in Artificial Intelligence*, págs. 29–38, Montreal, Agosto 1995.
- [DM98] Francisco Durán y José Meseguer. Arquitecturas de Software desde una Perspectiva Formal. Conferencia en JIS'98, Noviembre 1998.
- [DMC92] Christophe Dony, Jacques Malenfant y Pierre Cointe. Prototype-Based Languages: From a New Taxonomy to Constructive Proposals and Their Validation. En Andreas Paepcke, editor, *Proceedings of OOPSLA '92: Object-Oriented Programming Systems, Languages and Applications*, págs. 201–217, Vancouver, Canada, Octubre 1992. ACM, ACM Press. Also in SIGPLAN Notices 27(10).
- [DMSMTH96] Giovanna Di Marzo Serugendo, Murhimanya Muhugusa, Christian Tschudin y Jürgen Harms. Survey of Theories for Mobile Agents. *Cahier du CUI*, 106, Noviembre 1996. Centre Universitaire d'Informatique, University of Geneva.
- [DMT99] Grit Denker, José Meseguer y Carolyn Talcott. Rewriting Semantics of Distributed Meta Objects and Composable Communication Services, 1999.
- [DNR99] Elisabetta Di Nitto y David S. Rosenblum. Exploiting ADLs to Specify Architectural Styles Induced by Middleware Infrastructures. En *Proceedings of ICSE'21*, Mayo 1999.
- [Don99] Patrick Donohoe, editor. *Software Architecture – Proceedings of the First Working IFIP Conference on Software Architecture (WICSA1)*. Kluwer Academic Publishers, 1999.
- [dP99] Virgínia Carvalho Carneiro de Paula. *Implementação de Linguagens de Configuração para Sistemas Distribuídos*. Tesis Doctoral, Universidade Federal de Pernambuco, Recife, Brasil, 1999.
- [dPJC98] Virgínia Carvalho Carneiro de Paula, George Roger Ribeiro Justo y Paulo Roberto Freire Cunha. Specifying Dynamic Distributed Software Architectures. En *XII Simpósio Brasileiro de Engenharia de Software*. BCS Press, 1998.
- [DR97] Stéphane Ducasse y Tamar Richner. Executable Connectors: Towards Reusable Design Elements. En *Software Engineering – Proceedings of ESEC/FSE'97*, volumen 1301 de *Lecture Notes in Computer Science*, págs. 483–500, 1997.
- [dRS84] Jim des Rivières y Brian C. Smith. The Implementation of Procedurally Reflective Languages. En *Proceedings of ACM Symposium on Lisp and Functional Programming (L&FP'84)*, págs. 331–347, 1984.

- 
- [DS99] Rémi Douence y Mario Südholt. The Next 700 Reflective Object-Oriented Languages. Technical Report 99-1-INFO, École des Mines de Nantes, 1999.
- [DSC<sup>+</sup>00] Jim Dowling, Tilman Schäfer, Vinny Cahill, Peter Haraszti y Barry Redmond. Using Reflection to Support Dynamic Adaptation of System Software: A Case Study Driven Evaluation. En Walter Cazzola, Robert J. Stroud y Francesco Tisato, editores, *Reflection and Software Engineering*, volumen 1826 de *Lecture Notes in Computer Science*, págs. 171–190. Springer Verlag, Junio 2000.
- [Dul90] Naranker Dulay. *A Configuration Language for Distributed Programming*. Tesis Doctoral, Imperial College of Science, Technology and Medicine. University of London, Febrero 1990.
- [Dul92] Naranker Dulay. *darwin Language Reference Manual*. Technical report, Department of Computing, Imperial College, 1992.
- [EHIP<sup>+</sup>96] Walter J. Ellis, Richard F. Hilliard III, Peter T. Poon, David Rayford, Thomas F. Saunders, Basil Sherlund y Ronald L. Wade. Toward a Recommended Practice for Architectural Description. En *Proceedings of Second IEEE International Conference on Engineering of Complex Computer Systems*, Montreal, Quebec, Canadá, Octubre 1996. IEEE Architecture Planning Group.
- [EN00] Uffe H. Engberg y Mogens Nielsen. A Calculus of Communicating Systems with Label Passing – Ten Years After. En Plotkin et al. [PST00].
- [End94] Markus Endler. A Language for Implementing Generic Dynamic Reconfigurations of Distributed Programs. En *Proceedings of the 12<sup>th</sup> Brazilian Symposium on Computer Networks*, págs. 175–187, Curitiba, Mayo 1994.
- [Eug01] Patrick Thomas Eugster. *Type-Based Publish/Subscribe*. Tesis Doctoral, École Polytechnique Fédérale de Lausanne, Diciembre 2001.
- [EW92] Markus Endler y Jiawang Wei. Programming Generic Dynamic Reconfigurations for Distributed Applications. En *Proceedings of the 1<sup>st</sup> International Workshop on Configurable Distributed Systems*, págs. 68–79. IEE, 1992.
- [FA93] Svend Frølund y Gul Agha. A Language Framework for Multi-Object Coordination. En Nierstrasz [Nie93], págs. 346–360.
- [FA96] Svend Frølund y Gul A. Agha. Abstracting Interactions Based on Message Sets. En *Object-Based Models and Languages for Concurrent Systems*, volumen 924 de *Lecture Notes in Computer Science*, págs. 107–124. Springer-Verlag, 1996.
- [Fer89] Jacques Ferber. Computational Reflection in Class Based Object Oriented Languages. En Meyrowitz [Mey89], págs. 317–326.
- [FF96] Nissim Francez y Ira R. Forman. *Interacting Processes: A Multiparty Approach to Coordinated Distributed Programming*. Addison-Wesley, 1996.

- [FG95] Cédric Fournet y Georges Gonthier. The reflexive CHAM and the join-calculus. En *Proceedings of POPL'96*, Octubre 1995.
- [FGL<sup>+</sup>96] Cédric Fournet, Georges Gonthier, Jean-Jacques Lévy, Luc Maranget y Didier Rémy. A Calculus of Mobile Agents. En *Proceedings of CONCUR'96*, 1996.
- [FHA99] Eric Freeman, Susanne Hupfer y Ken Arnold. *JavaSpaces Principles, Patterns, and Practice*. Jini Technology Series. Addison-Wesley Publishing Co., Junio 1999.
- [Fil94] Andrzej Filinski. Representing Monads. En *Proceedings of ACM POPL'94*, págs. 446–457, 1994.
- [FLMR00] Cédric Fournet, Cosimo Laneve, Luc Maranget y Didier Rémy. Inheritance in the Join Calculus. En *Foundations of Software Technology and Theoretical Computer Science*, volumen 1974 de *Lecture Notes in Computer Science*. Springer Verlag, Diciembre 2000.
- [FM96] José Luiz Fiadeiro y Tom S.E. Maibaum. A Mathematical Toolbox for the Software Architect. En Jeff Kramer y Alexander Wolf, editores, *Proceedings 8<sup>th</sup> International Workshop on Software Specification and Design*, págs. 46–55. IEEE Computer Society Press, 1996.
- [FM97] José Luiz Fiadeiro y Tom S. E. Maibaum. Categorical Semantics of Parallel Program Design. *Science of Computer Programming*, 28:111–138, 1997.
- [FM98] Pascal Fradet y Daniel Le Métayer. Structured Gamma. *Science of Computer Programming*, 31(2–3):263–289, Julio 1998.
- [FML55] José Ferrater Mora y Hugues Leblanc. *Lógica Matemática*. Fondo de Cultura Económica, México, 1955.
- [Foo89] Brian Foote. Class Warfare: Classes vs. Prototypes. En David Ungar, editor, *OOPSLA'89 Workshop on Objects without Classes*, New Orleans, Louisiana, Agosto 1989. ACM.
- [For98] Thomas Forster. Quine's NF – 60 Years On. *American Mathematical Monthly*, Julio 1998.
- [For00] Ira R. Forman. Superimposition: A Form of Separation of Concerns for Distributed Systems. En *Electronic Proceedings of First OOPSLA Workshop on Advanced Separation of Concerns (ASoC'00)*, Octubre 2000.
- [Fos97] Halldor Fosså. *Interactive Configuration Management for Distributed Systems*. Tesis Doctoral, Imperial College of Science, Technology and Medicine. University of London, Abril 1997.

- 
- [FW84] Daniel P. Friedman y Mitchell Wand. Reification: Reflection Without Metaphysics. En *Proceedings of ACM Symposium on Lisp and Functional Programming*, págs. 348–355, 1984.
- [GAO94] David Garlan, Robert Allen y John Ockerbloom. Exploiting Style in Architectural Design Environments. En *Proceedings of SIGSOFT: Foundations of Software Engineering (FSE'94)*, págs. 175–188, Diciembre 1994.
- [GAO95] David Garlan, Robert Allen y John Ockerbloom. Architectural Mismatch, or, Why It's Hard to Build Systems Out of Existing Parts. En *Proceedings of ICSE'17*, Seattle, Washington, Abril 1995.
- [Gar95] David Garlan. What is style? En *Proceedings of IWASS'1*, Abril 1995.
- [Gar98] David Garlan. Higher-Order Connectors. En *Workshop on Compositional Software Architectures*, Enero 1998.
- [Gar00] David Garlan. Software Architecture: a Roadmap. En Anthony Finkelstein y Jeff Kramer, editores, *The Future of Software Engineering*, págs. 91–101. ACM Press, Limerick, Ireland, Junio 2000. Proceedings of 22<sup>nd</sup> International Conference on Software Engineering (ICSE 2000), Vol. 2.
- [Gay99] Simon Gay. Some Type Systems for the Pi Calculus. Technical report, Department of Computer Science, Royal Holloway, University of London, 1999.
- [GC92] David Gelernter y Nicholas Carriero. Coordination languages and their significance. *Communications of the ACM*, 35(2):97–107, Febrero 1992.
- [GC96] Brendan Gowing y Vinny Cahill. Meta-Object Protocols for C++: The Iguana Approach. En *Proceedings of Reflection 96*, págs. 137–152, 1996.
- [Göd31] Kurt Gödel. Über formal unentscheidbare Sätze der *Principia Mathematica* und verwandter Systeme I. *Monatshefte für Mathematik und Physik*, 38:173–198, 1931.
- [Gel82] David Gelernter. *An Integrated Microcomputer Network for Experiments in Distributed Programming*. Tesis Doctoral, State University of New York at Stony Brook, Octubre 1982.
- [Gel85] David Gelernter. Generative Communication in Linda. *ACM Transactions on Programming Languages and Systems*, 7(1):80–112, Enero 1985.
- [Ger96] Capt. Mark Gerken. Minutes of the First EDCS Architecture Cluster Workshop. Napa Valley, DARPA/SEI/RL, Octubre 1996.
- [GHJ<sup>+</sup>95a] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides y Grady Booch. *Design Patterns*. Addison-Wesley, 1995.

- [GHJ<sup>+</sup>95b] Erich Gamma, Richard Helm, Ralph E. Johnson, John Vlissides y Grady Booch. *Design Patterns*. Addison-Wesley, 1995.
- [GHJV94] Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, Octubre 1994.
- [Gia99] Dimitra Giannakopoulou. *Model Checking for Concurrent Software Architectures*. Tesis Doctoral, Imperial College of Science, Technology and Medicine. University of London, Enero 1999.
- [GK96] Kaveh Moazami Goudarzi y Jeff Kramer. Maintaining Node Consistency in the Face of Dynamic Change. En *Proceedings of the Third International Conference on Configurable Distributed Systems*, págs. 62–69. IEEE Computer Society Press, Mayo 1996.
- [GLM97] David Garlan y Daniel Le Métayer, editores. *Second International Conference on Coordination Languages and Models (Coordination'97)*, volumen 1282 de *Lecture Notes in Computer Science*, Berlin, Septiembre 1997. Springer Verlag.
- [GMW97] David Garlan, Robert T. Monroe y David Wile. ACME: An Architectural Interchange Language. En *Proceedings of CASCON'97*, Noviembre 1997.
- [GN91] David Garlan y David Notkin. Formalizing Design Spaces: Implicit Invocation Mechanisms. En *Proceedings of VDM'91: Formal Software Development Methods*, volumen 551 de *Lecture Notes in Computer Science*. Springer-Verlag, Octubre 1991.
- [Gol97] Michael Golm. Design and Implementation of a Meta Architecture for Java. Tesis de Maestría, Institut für Mathematische und Datenverarbeitung der Friedrich-Alexander-Universität Erlangen-Nürnberg, Enero 1997.
- [Gor00] Andrew D. Gordon. Notes on Nominal Calculi for Security and Mobility. Lecture Notes for FOSAD'00 Summer School, Julio 2000.
- [GP95] David Garlan y Dewayne E. Perry. Introduction to the Special Issue on Software Architecture. *IEEE Transactions on Software Engineering*, 21(4), Abril 1995.
- [GQ94] Michael M. Gorlick y Alex Quilici. Visual Programming in the Large versus Visual Programming in the Small. En *Proceedings of 10<sup>th</sup> IEEE Conference on Visual Languages*, págs. 137–144, St. Louis, Noviembre 1994.
- [Gör90] Steven Kent Göring. *A Graph Grammar Approach to Concurrent Programming – Delta Grammars*. Tesis Doctoral, University of Illinois, Department of Computer Science, 1990. UIUC-DCS-R-90-1576.
- [GS93] David Garlan y Mary Shaw. An Introduction to Software Architecture. Technical Report CMU/SEI-94-TR-21, Software Engineering Institute, Carnegie Mellon University, 1993.

- [GW00] Philippa Gardner y Lucian Wischik. Explicit Fusions. En Mogens Nielsen y Branislav Rovan, editores, *25<sup>th</sup> International Symposium on the Mathematical Foundations of Computer Science (MFCS 2000)*, volumen 1893 de *Lecture Notes in Computer Science*, págs. 373–382, Bratislava, Septiembre 2000. Springer.
- [Har95] John Harrison. Metatheory and Reflection in Theorem Proving: A Survey and Critique. Technical Report SRI-TR CRC-053, University of Cambridge, Computer Laboratory, SRI, Febrero 1995.
- [Hew77] Carl Hewitt. Viewing control structures as patterns of passing messages. *Journal of Artificial Intelligence*, 8(3), 1977.
- [HIM99] Dan Francisco Hirsch, Paola Inverardi y Ugo Montanari. Modelling Software Architectures and Styles with Graph Grammars and Constraint Solving. En Donohoe [Don99], págs. 127–143.
- [Höl93] Urs Hölzle. Integrating Independently-Developed Components in Object-Oriented Languages. En Nierstrasz [Nie93], págs. 36–56.
- [HL95] Walter Hürsch y Cristina Videira Lopes. Separation of Concerns. Technical Report NU-CCS-95-03, Northeastern University, Febrero 1995.
- [HLMS92] Chris Hankin, Daniel Le Métayer y David Sands. A Calculus of Gamma Programs. En *Fifth Workshop on Languages and Compilers for Parallel Computing*, volumen 757 de *Lecture Notes in Computer Science*, Yale, 1992. Springer Verlag.
- [HM94] Richard C. Holt y Spiros Mancoridis. Using Tube Graphs to Model Architectural Designs of Software Systems. Technical Report 308, University of Toronto, CSRI, Octubre 1994.
- [HM99] Dan Francisco Hirsch y Ugo Montanari. Consistent Transformations for Software Architecture Styles of Distributed Systems. En Gheorghe Stefanescu, editor, *Workshop on Distributed Systems*, volumen 28 de *Electronic Notes in Theoretical Computer Science*. Elsevier, 1999.
- [HNS00] Christine Hofmeister, Robert Nord y Dilip Soni. *Applied Software Architecture*. The Object Technology Series. Addison-Wesley Longman, 2000.
- [Hoa85] C. Anthony R. Hoare. *Communicating Sequential Processes*. Prentice-Hall, 1985.
- [Hol96a] Ric C. Holt. Binary Relational Algebra Applied to Software Architecture. Technical Report CSRI TR-345, University of Toronto, Junio 1996.
- [Hol96b] A. A. Holzbacher. A Software Environment for Concurrent Coordinated Programming. En Ciancarini y Hankin [CH96], págs. 249–266.
- [Hol98] Richard C. Holt. Structural Manipulations of Software Architecture Using Tarski Relational Algebra. En *5<sup>th</sup> Working Conference on Reverse Engineering*, Octubre 1998.

- [Hol99] David Holmes. *Synchronisation Rings: Composable Synchronization for Object-Oriented Systems*. Tesis Doctoral, Department of Computing, Macquarie University, Sydney, Australia, 1999.
- [HP93] Christine Hofmeister y James M. Purtilo. Dynamic Reconfiguration in Distributed Systems: Adapting Software Modules for Replacement. En *Proceedings of 13<sup>th</sup> International Conference on Distributed Computing Systems*, págs. 101–110. IEEE CS Press, Mayo 1993.
- [HT91] Kohei Honda y Mario Tokoro. An Object Calculus for Asynchronous Communication. En Pierre America, editor, *ECOOOP'91: European Conference on Object-Oriented Programming*, volumen 512 de *Lecture Notes in Computer Science*, págs. 133–147, Geneve, Switzerland, Julio 1991. Springer-Verlag.
- [Hun81] Geoffrey Hunter. *Metalógica: Introducción a la Metateoría de la Lógica Clásica de Primer Orden*. Lógica y Teoría de la Ciencia. Paraninfo, 1981.
- [HVK98] Kohei Honda, Vasco Thudichum Vasconcelos y Makoto Kubo. Language Primitives and Type Discipline for Structured Communication-based Programming. En Chris Hankin, editor, *Programming Languages and Systems: Proceedings of the 7<sup>th</sup> European Symposium on Programming (ESOP'98)*, volumen 1381 de *Lecture Notes in Computer Science*, págs. 122–138. Springer Verlag, Abril 1998.
- [IB96] Valérie Issarny y Christophe Bidan. Aster: A Framework for Sound Customization of Distributed Runtime Systems. En *Proceedings of ICDCS'96*, Mayo 1996.
- [IdFF96] Roberto Ierusalimsky, Luiz Henrique de Figueiredo y Waldemar Celes Filho. Lua – an Extensible Extension Language. *Software: Practice and Experience*, 26(6):635–652, 1996.
- [IW95] Paola Inverardi y Alexander L. Wolf. Formal Specification and Analysis of Software Architectures using the Chemical Abstract Machine Model. *IEEE Transactions on Software Engineering*, 21(4):373–386, Abril 1995.
- [IWY97] Paola Inverardi, Alexander L. Wolf y Daniel Yankelevich. Checking Assumptions in Component Dynamics at the Architecture Level. En Garlan y Le Métayer [GLM97], págs. 46–63.
- [IWY98] Paola Inverardi, Alexander L. Wolf y Daniel Yankelevich. Behavioural Type Checking of Architectural Components Based on Assumptions. Technical Report CU-CS-861-98, University of Colorado at Boulder, Abril 1998.
- [IWY00] Paola Inverardi, Alexander L. Wolf y Daniel Yankelevich. Static Checking of System Behaviors Using Derived Component Assumptions. *ACM Transactions on Software Engineering and Methodology*, 2000.

- 
- [Jac98] Bart Jacobs. Coalgebraic Reasoning About Classes in Object-Oriented Languages. *Electronic Notes in Theoretical Computer Science*, 11:1–12, 1998.
- [JC92] George Roger Ribeiro Justo y Paulo Roberto Freire Cunha. Programming Distributed Systems with Configuration Languages. En *Proceedings of the 1<sup>st</sup> International Workshop on Configurable Distributed Systems*, págs. 118–127. IEE, Marzo 1992.
- [JF90] Ramón Jansana Ferrer. *Una Introducción a la Lógica Modal*. Cuadernos de Filosofía y Ensayo. Tecnos, 1990.
- [Jon96] Cliff B. Jones. Accommodating Interference in the Formal Design of Concurrent Object-Based Programs. *Formal Methods in System Design*, 8(2):105–122, Marzo 1996.
- [JR97] Bart Jacobs y Jan J.M.M. Rutten. A Tutorial on (Co)Algebras and (Co)Induction. *Bulletin of EATCS*, 62:222–259, 1997.
- [JRvdL00] Mehdi Jazayeri, Alexander Ran y Frank van der Linden, editores. *Software Architecture for Product Families: Principles and Practice*. Addison-Wesley, 2000.
- [JS96] Yuh-Jzer Joung y Scott A. Smolka. A Comprehensive Study of the Complexity of Multiparty Interaction. *Journal of the ACM*, 43(1):75–115, 1996.
- [Kam92] Fairouz Kamareddine. Set Theory and Nominalisation, Part I. *Journal of Logic and Computation*, 2(5):579–604, 1992.
- [Kat93] Shmuel Katz. A Superimposition Control Construct for Distributed Systems. *ACM Transactions on Programming Languages and Systems*, 15(2):337–356, Abril 1993.
- [Ken95] John J. Kenney. *Executable Formal Models of Distributed Transaction Systems based on Event Processing*. Tesis Doctoral, University of Stanford, CSL, Diciembre 1995.
- [KFRGC98a] Marc-Olivier Killijian, Jean-Charles Fabre, Juan-Carlos Ruiz-García y Shigeru Chiba. A Metaobject Protocol for Fault-Tolerant CORBA Applications. En *Proceedings of the 17<sup>th</sup> IEEE Symposium on Reliable Distributed Systems (SRDS'98)*, págs. 127–134. IEEE Press, 1998.
- [KFRGC98b] Marc-Olivier Killijian, Jean-Charles Fabre, Juan-Carlos Ruiz-García y Shigeru Chiba. A Metaobject Protocol for Fault-Tolerant CORBA Applications. Technical Report RR-98-139, Laboratoire d'Analyse et d'Architecture des Systèmes, Centre National de la Recherche Scientifique (LAAS-CNRS), Toulouse-Cedex, France, Abril 1998.

- [KG99] Shmuel Katz y Yossi Gil. Aspects and Superimpositions. En *Electronic Proceedings of Third ECOOP Workshop on Aspect-Oriented Programming (AOP'99)*, Junio 1999.
- [KGO<sup>+</sup>00] Rohit Khare, Michael Gunter, Peyman Oreizy, Nenad Medvidovic y Richard N. Taylor. xADL: Enabling Architecture-Centric Tool Integration With XML. Technical Report UCI-ICS-00-12, Department of Information and Computer Science, University of California, Irvine, Abril 2000.
- [KHL<sup>+</sup>93] Gail E. Kaiser, Wenwey Hseush, James C. Lee, Shyhtsun F. Wu, Esther Woo, Eric Hilsdale y Scott Meyer. MELDC: A Reflective Object-Oriented Coordination Language. Technical Report CU-CS-001-93, Department of Computer Science, Columbia University, Nueva York, Enero 1993.
- [Kir92] Graham N. C. Kirby. *Reflection and Hyper-Programming in Persistent Programming Systems*. Tesis Doctoral, University of St. Andrews, Escocia, 1992. Department of Mathematical and Computational Sciences.
- [KKVvV01] Rick Kazman, Philippe Kruchten, Chris Verhoef y Hans van Vliet, editores. *Proceedings of the Second Working IEEE/IFIP Conference on Software Architecture (WICSA'01)*. IEEE Computer Society Press, Amsterdam, Agosto 2001.
- [KL01] Fairouz Kamareddine y Twan Laan. A Correspondence between Martin-Löf Type Theory, the Ramified Theory of Types and Pure Type Systems. *Journal of Logic, Language and Information*, 10(3):375–402, 2001. Kluwer Academic Publishers.
- [Kle74] Stephen Cole Kleene. *Introducción a la Metamatemática*. Estructura y Función. Tecnos, 1974.
- [KLM94] Dinesh Katiyar, David C. Luckham y John Mitchell. A type system for prototyping languages. En *Proceedings of 21<sup>st</sup> ACM Symposium on Principles of Programming Languages (PoPL'94)*, Portland, Oregón, 1994. ACM Press.
- [KLM<sup>+</sup>97] Gregor Kiczales, John Lamping, Anurag Mendhekar, Chris Maeda, Cristina Videira Lopes, Jean-Marc Loingtier y John Irwin. Aspect-Oriented Programming. En Mehmet Aksit y Satoshi Matsuoka, editores, *Proceedings of the European Conference on Object-Oriented Programming (ECOOP'97)*, número 1241 en Lecture Notes in Computer Science, págs. 220–242, Jyvaskyla, Finland, Junio 1997. Springer Verlag.
- [KM85] Jeff Kramer y Jeffrey N. Magee. Dynamic Configuration for Distributed Systems. *IEEE Transactions on Software Engineering*, 11(4):424–436, 1985.
- [KM90] Jeff Kramer y Jeff Magee. The Evolving Philosophers Problem: Dynamic Change Management. *IEEE Transactions on Software Engineering*, 16, Noviembre 1990.

- 
- [KM96a] Rudolf Kaehr y Thomas Mahler. Introducing and Modeling Polycontextual Logics. En *13<sup>th</sup> European Meeting on Cybernetics and Systems Research*, Abril 1996.
- [KM96b] Jeff Kramer y Jeff Magee. Self Organising Software Architectures. En L. Vidal, A. Finkelstein, G. Spanoudakis y Alexander L. Wolf, editores, *Joint Proceedings of the SIGSOFT'96 Workshops*, págs. 35–38. ACM Press, 1996.
- [KMS98] Graham Kirby, Ron Morrison y David Stemple. Linguistic Reflection in Java. Technical Report PL/9810027, University of St. Andrews, Scotland, Octubre 1998.
- [Kru95] Philippe Kruchten. The 4+1 View Model of Architecture. *IEEE Software*, 12(6), Noviembre 1995.
- [Kru99] Philippe Kruchten. *The Rational Unified Process - An Introduction*. Object Technology Series. Addison-Wesley, 1999.
- [LAN00] Markus Lumpe, Franz Achermann y Oscar M. Nierstrasz. A Formal Language for Composition. En Gary T. Leavens y Murali Sitaraman, editores, *Foundations of Component-Based Systems*, págs. 69–90. Cambridge University Press, 2000.
- [Lea94] Doug Lea. Christopher Alexander: An Introduction for Object-Oriented Designers. *ACM Software Engineering Notes*, Enero 1994.
- [Led98] Thomas Ledoux. *Réflexion dans les Systèmes Répartis: application à CORBA et SmallTalk*. Tesis Doctoral, Université de Nantes, École des Mines de Nantes, Marzo 1998.
- [Led99] Thomas Ledoux. OpenCorba: a Reflective Open Broker. En Pierre Cointe, editor, *Proceedings of the Second International Conference on Reflection (Reflection'99)*, volumen 1616 de *Lecture Notes in Computer Science*, págs. 197–214, Saint Malo, France, Julio 1999. Springer Verlag.
- [LKA<sup>+</sup>95] David C. Luckham, John J. Kenney, Larry M. Augustin, James Vera, Doug Bryan y Walter Mann. Specification and Analysis of System Architecture using Rapide. *IEEE Transactions on Software Engineering*, 21(4):336–355, Abril 1995.
- [LM94] Daniel Le Métayer. Higher-Order Multiset Programming. En *DIMACS Workshop on Specifications of Parallel Algorithms*, volumen 18 de *DIMACS Series in Discrete Mathematics*. American Mathematical Society, 1994.
- [LM96] Daniel Le Métayer. Software Architecture Styles As Graph Grammars. En *Proceedings of the 4<sup>th</sup> ACM SIGSOFT Symposium on the Foundations of Software Engineering*, págs. 15–24. ACM Press, 1996.
- [LM98] Daniel Le Métayer. Describing Software Architecture Styles Using Graph Grammars. *IEEE Transactions on Software Engineering*, 24(7):521–553, Julio 1998.

- [LP91] Wilf LaLonde y John Pugh. Subclassing  $\neq$  Subtyping  $\neq$  Is-a. *Journal of Object-Oriented Programming*, 3(5):57–62, Enero 1991.
- [LSLL00] Orlando Loques, Alexandre Sztajnberg, Julius Leite y Marcelo Lobosco. On the Integration of Configuration and Meta-Level Programming Approaches. En Cazzola et al. [CST00], págs. 191–210.
- [LSN96] Markus Lumpe, Jean-Guy Schneider y Oscar Marius Nierstrasz. Using Metaobjects to Model Concurrent Objects with Pict. En *Proceedings of Languages et Modèles à Objets*, págs. 1–12, Leysin, Suiza, Octubre 1996.
- [Luc97] Carine Lucas. *Documenting Reuse and Evolution with Reuse Contracts*. Tesis Doctoral, Vrije Universiteit Brussel, Septiembre 1997.
- [Lum99] Markus Lumpe. *A  $\pi$ -Calculus Based Approach for Software Composition*. Tesis Doctoral, Institut für Informatik und angewandte Mathematik, Universität Bern, Switzerland, Enero 1999.
- [LV95] David C. Luckham y James Vera. An Event-Based Architecture Definition Language. *IEEE Transactions on Software Engineering*, 21(9):717–734, Septiembre 1995.
- [LVB<sup>+</sup>93] David C. Luckham, James Vera, Doug Bryan, Larry Augustin y Frank Belz. Partial Orderings of Event Sets and Their Application to Prototyping Concurrent, Timed Systems. *Journal of Systems and Software*, 21(3):253–265, Junio 1993.
- [LVM95] David C. Luckham, James Vera y Sigurd Meldal. Three Concepts of System Architecture. Technical Report CSL-TR-95-674, University of Stanford, CSL, 1995.
- [LW95] Xinxin Liu y David Walker. A Polymorphic Type System for the Polyadic  $\pi$ -calculus. En Insup Lee y Scott A. Smolka, editores, *Proceedings of Sixth International Conferencia on Concurrency Theory (CONCUR 1995)*, volumen 962 de *Lecture Notes in Computer Science*, págs. 103–116, Philadelphia, Agosto 1995. Springer.
- [LW00] Xinxin Liu y David Walker. Concurrent Objects as Mobile Processes. En Plotkin et al. [PST00].
- [LWF01] Antónia Lopes, Michel Wermelinger y José Luiz Fiadeiro. Higher-Order Architectural Connectors. Technical report, Universidade de Lisboa, 2001.
- [Lyn96] Nancy A. Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1996.
- [Mae87a] Pattie Maes. Concepts and Experiments in Computational Reflection. En Meyerowitz [Mey87], págs. 147–155.
- [Mae87b] Pattie Maes. *Concepts and Experiments in Computational Reflection*. Tesis Doctoral, Vrije Universiteit Brussel, 1987.

- [Mar99] Eric Marsden. Description Formelle d'un Protocole à Métaobjets. Tesis de Maestría, Laboratoire d'Analyse et d'Architecture des Systèmes, Centre National de la Recherche Scientifique (LAAS-CNRS), Toulouse-Cedex, France, Junio 1999.
- [Mar01] Claudia A. Marcos. *Patrones de Diseño como Entidades de Primera Clase*. Tesis Doctoral, Universidad Nacional del Centro de la Provincia de Buenos Aires, Instituto de Sistemas ISISTAN, Tandil, Argentina, Abril 2001.
- [Mas99] Cecilia Mascolo. MobiS: A Specification Language for Mobile Systems. En Ciancarini y Wolf [CW99], págs. 37–52.
- [MC94] Thomas W. Malone y Kevin Crowston. The Interdisciplinary Study of Coordination. *ACM Computing Surveys*, 26(1):87–119, Marzo 1994.
- [McA95] Jeff McAffer. Meta-Level Programming with CodA. En Walter Olthoff, editor, *Proceedings of the 9<sup>th</sup> European Conference on Object-Oriented Programming (ECOOP'95)*, volumen 952 de *Lecture Notes in Computer Science*, págs. 190–214. Springer Verlag, 1995.
- [McA96] Jeff McAffer. Engineering the Meta-Level. En *Proceedings of Reflection'96*, págs. 39–61, Abril 1996.
- [McH94] Ciaran McHale. *Synchronisation in Concurrent, Object-Oriented Languages: Expressive Power, Genericity and Inheritance*. Tesis Doctoral, Trinity College, University of Dublin, Octubre 1994.
- [McI69] M. Douglas McIlroy. Mass-Produced Software Components. En Peter Naur y Brian Randell, editores, *Software Engineering: Report on a Conference Sponsored by the NATO Science Committee*, págs. 138–150, Garmisch, Alemania, Octubre 1969.
- [MDEK95] Jeff N. Magee, Naranker Dulay, Susan Eisenbach y Jeff Kramer. Specifying Distributed Software Architectures. En *Proceedings of the Fifth European Software Engineering Conference (ESEC'95)*, Septiembre 1995.
- [Med96a] Nenad Medvidovic. ADLs and Dynamic Architecture Changes. En *Proceedings of the Second International Software Architecture Workshop (ISAW'2/SIGSOFT'96)*, págs. 24–27, San Francisco, CA, 1996.
- [Med96b] Neno Medvidovic. A Classification and Comparison Framework for Software Architecture Description Languages. Technical Report UCI-ICS-97-02, University of California Irvine, Febrero 1996.
- [Med99] Nenad Medvidovic. *Architecture-Based Specification-Time Software Evolution*. Tesis Doctoral, University of California Irvine, 1999.
- [Mei95] Theo Dirk Meijler. Viewing Software Composition as Composition of Metaobjects. Technical report, University of Bern, 1995.

- [Mes90] José Meseguer. A Logical Theory of Concurrent Objects. En *Proceedings of ECOOP-OOPSLA '90 Conference on Object-Oriented Programming*, págs. 101–115, Ottawa, Canadá, Octubre 1990. ACM Press.
- [Mes92] José Meseguer. Conditional Rewriting Logic as a Unified Model of Concurrency. *Theoretical Computer Science*, 96:73–155, 1992.
- [Mes96] José Meseguer. Rewriting Logic as a Semantic Framework for Concurrency: A Progress Report. En *Proceedings of CONCUR '96*, número 1119 en Lecture Notes in Computer Science, págs. 331–372. Springer Verlag, 1996.
- [Mey87] Norman Meyrowitz, editor. *OOPSLA '87 Conference Proceedings*, volumen 22(12) de *SIGPLAN Notices*. ACM Press, Diciembre 1987.
- [Mey89] Norman Meyrowitz, editor. *OOPSLA '89 Conference Proceedings*, volumen 24(10) de *SIGPLAN Notices*. ACM Press, Octubre 1989.
- [MG99] Kaveh Moazami-Goudarzi. *Consistency-Preserving Dynamic Reconfiguration of Distributed Systems*. Tesis Doctoral, Imperial College of Science, Technology and Medicine. University of London, Marzo 1999.
- [MHSA98] Juan M. Murillo, Juan Hernández, Fernando Sánchez y Luis A. Alvarez. Roles y Coordinadores de Roles: Componentes de Coordinación para Objetos Activos. En *Actas de las III Jornadas de Ingeniería del Software*, págs. 231–242, Murcia, Noviembre 1998. Diego Marín.
- [Mil80] A. J. Robin G. Milner. *A Calculus of Communicating Systems*, volumen 92 de *Lecture Notes in Computer Science*. Springer Verlag, 1980.
- [Mil89] Arthur J. Robin G. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.
- [Mil92] Robin Milner. Functions as Processes. *Mathematical Structures in Computer Science*, 2(2):119–141, 1992.
- [Mil93a] Robin Milner. Elements of Interaction. *Communications of the ACM*, 36(1):78–89, Enero 1993. Turing Award Lecture.
- [Mil93b] Robin Milner. The Polyadic  $\pi$ -calculus: A Tutorial. En F.L. Bauer, W. Brauer y H. Schwichtenberg, editores, *Logic and Algebra of Specification*, págs. 203–246. Springer Verlag, 1993.
- [Mil96] Robin Milner. Calculi for Interaction. *Acta Informatica*, 33:707–737, 1996.
- [Mil99] Robin Milner. *Communicating and Mobile Systems: the Pi-Calculus*. Cambridge University Press, Junio 1999.

- [Mil01] Robin Milner. Bigraphical Reactive Systems. En Kim Guldstrand Larsen y Mogens Nielsen, editores, *CONCUR 2001 - Proceedings of 12<sup>th</sup> International Conference on Concurrency Theory*, volumen 2154 de *Lecture Notes in Computer Science*, págs. 16–35, Aalborg, Denmark, Agosto 2001. Springer.
- [Min97] Naftaly Minsky. Towards Architectural Invariants of Evolving Systems. Technical report, Rutgers University, New Brunswick, NJ, Febrero 1997.
- [MJD96] Jacques Malenfant, Marco Jacques y François-Nicola Demers. A Tutorial on Behavioural Reflection and its Implementation. En *Proceedings of Reflection 96*, págs. 1–20, 1996.
- [MK96] Jeff Magee y Jeff Kramer. Dynamic Structure in Software Architectures. *ACM Software Engineering Notes*, 21(6):3–14, Noviembre 1996.
- [MKG97] Jeff Magee, Jeff Kramer y Dimitra Giannakopoulou. Analysing the Behaviour of Distributed Software Architectures: a Case Study. En *5<sup>th</sup> IEEE Workshop on Future Trends of Distributed Computing Systems*, Octubre 1997.
- [MKG99] Jeff Magee, Jeff Kramer y Dimitra Giannakopoulou. Behaviour Analysis of Software Architectures. En *Software Architecture*, págs. 35–49. Kluwer Academic Publishing, Febrero 1999.
- [MKS89] Jeff Magee, Jeff Kramer y Morris Sloman. Constructing Distributed Systems in Conic. *IEEE Transactions on Software Engineering*, 15(6), 1989.
- [MKS94] Jeff Magee, Jeff Kramer y Morris Sloman. Regis: A Constructive Development Environment for Distributed Programs. *Distributed Systems Engineering*, 1(5):663–675, 1994.
- [MM00] Raphael Malveau y Thomas Mowbray. *The Software Architect Bootcamp*. Prentice-Hall, Octubre 2000.
- [MMP00] Nikunj R. Mehta, Nenad Medvidovic y Sandeep Phadke. Towards a Taxonomy of Software Connectors. En *Proceedings of the 22<sup>nd</sup> International Conference on Software Engineering (ICSE 2000)*, Junio 2000.
- [MN88] Pattie Maes y Daniele Nardi, editores. *Meta-Level Architectures and Reflection*. Elsevier Science – North-Holland, Febrero 1988.
- [MN97] Theo Dirk Meijler y Oscar Marius Nierstrasz. Beyond Objects: Components. En M.P. Papazoglou y G. Schlageter, editores, *Cooperative Information Systems: Current Trends and Directions*, págs. 49–78. Academic Press, Noviembre 1997.
- [MNS95] Gail C. Murphy, David Notkin y Kevin Sullivan. Software Reflexion Models: Bridging the Gap between Source and High-Level Models. En *Proceedings of the Third ACM Symposium on the Foundations of Software Engineering (SIGSOFT-FSE'95)*, volumen 20 de *ACM Software Engineering Notes*, págs. 18–28, Washington, DC, Octubre 1995. ACM Press.

- [MOM96] Narciso Martí-Oliet y José Meseguer. Rewriting Logic as a Logical and Semantic Framework. En *Proceedings of First International Workshop on Rewriting Logic and its Applications*, volumen 4 de *Electronic Notes in Theoretical Computer Science*, Septiembre 1996.
- [Mon98] Robert T. Monroe. Capturing Software Architecture Design Expertise With Armani. Technical Report CMU-CS-98-163, Carnegie Mellon University School of Computer Science, Octubre 1998.
- [MPW92a] Robin Milner, Joachim Parrow y David Walker. A Calculus of Mobile Processes, Part I. *Information and Computation*, 100:1–40, 1992.
- [MPW92b] Robin Milner, Joachim Parrow y David Walker. A Calculus of Mobile Processes, Part II. *Information and Computation*, 100:41–77, 1992.
- [MR87] Naftaly H. Minsky y David Rozenshtein. A Law-Based Approach to Object-Oriented Programming. En Meyrowitz [Mey87], págs. 482–493.
- [MR89] Naftaly H. Minsky y David Rozenshtein. Controllable Delegation: An Exercise in Law-Governed Systems. En Meyrowitz [Mey89], págs. 371–380.
- [MR99] Nenad Medvidovic y David S. Rosenblum. Assessing the Suitability of a Standard Design Method for Modeling Software Architectures. En *Software Architecture*, págs. 161–182. Kluwer Academic Publishing, Febrero 1999.
- [MRT98] Nenad Medvidovic, David S. Rosenblum y Richard N. Taylor. A Type Theory for Software Architectures. Technical Report UCI-ICS-98-04, University of California Irvine, Febrero 1998.
- [MRT99] Nenad Medvidovic, David S. Rosenblum y Richard N. Taylor. A Language and Environment for Architecture-Based Software Development and Evolution. En *Proceedings of ICSE'21*, Mayo 1999.
- [MT97a] Nenad Medvidovic y Richard N. Taylor. A Framework for Classifying and Comparing Architecture Description Languages. En *Software Engineering – Proceedings of ESEC/FSE'97*, págs. 60–76. ACM Press, Septiembre 1997.
- [MT97b] José Meseguer y Carolyn Talcott. Semantic Interoperation of Dynamic Heterogeneous Architectures. Part I: A Case Study in Semantic Interoperation of Dynamic Heterogeneous Architectures. Seattle EDCS Workshop, Julio 1997.
- [MT97c] José Meseguer y Carolyn Talcott. Semantic Interoperation of Dynamic Heterogeneous Architectures. Part I: Rewriting Logic as a Semantic Framework for Formal Interoperation of Architecture Styles and Languages. Santa Fe EDCS Architecture Cluster Meeting, Abril 1997.
- [MT98] Nenad Medvidovic y Richard N. Taylor. Separating Fact from Fiction in Software Architecture. En *Proceedings of ISAW'3*, págs. 105–108. ACM Press, 1998.

- 
- [MT00] Nenad Medvidovic y Richard N. Taylor. A Classification and Comparison Framework for Software Architecture Description Languages. *IEEE Transactions on Software Engineering*, 26(1):70–93, Enero 2000.
- [MTW96] Nenad Medvidovic, Richard N. Taylor y E. James Whitehead, Jr. Formal Modeling of Software Architectures at Multiple Levels of Abstraction. En *Proceedings of California Symposium 1996*, págs. 28–40, Abril 1996.
- [MU97] Naftaly Minsky y Victoria Ungureanu. Regulated Coordination in Open Distributed Systems. En Garlan y Le Métayer [[GLM97](#)].
- [MU99] Naftaly Minsky y Victoria Ungureanu. Law-Governed Interaction: A Coordination & Control Mechanism for Heterogeneous Distributed Systems. Draft, 1999.
- [MU00] Naftaly Minsky y Victoria Ungureanu. Law-Governed Interaction: A Coordination & Control Mechanism for Heterogeneous Distributed Systems. *ACM Transactions on Software Engineering and Methodology*, 9(3):273–305, Julio 2000.
- [MUWZ96] Naftaly Minsky, Victoria Ungureanu, Wenhui Wang y Junbiao Zhang. Building Reconfiguration Primitives into the Law of a System. En *International Conference on Configurable Distributed Systems (ICCDs'96)*, Mayo 1996.
- [NACO97] Gleb Naumovich, George S. Avrunin, Lori A. Clarke y Leon J. Osterweil. Applying Static Analysis to Software Architectures. En *Proceedings of the 6<sup>th</sup> European Software Engineering Conference and 5<sup>th</sup> ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE 97)*, volumen 1301 de *Lecture Notes in Computer Science*, págs. 77–93, Zurich, Septiembre 1997. Springer Verlag.
- [Ng92] Keng Ng. *Visual Support for Distributed Programming*. Tesis Doctoral, Imperial College of Science, Technology and Medicine. University of London, 1992.
- [Nie93] Oscar Marius Nierstrasz, editor. *ECOOP'93 – Object-Oriented Programming*, volumen 707 de *Lecture Notes in Computer Science*. Springer-Verlag, Julio 1993.
- [NY93] Shojiro Nishio y Akinori Yonezawa, editores. *Object Technologies for Advanced Software*, volumen 742 de *Lecture Notes in Computer Science*, Kanazawa, Japón, Noviembre 1993. Springer Verlag.
- [OB98] Alexandre Oliva y Luiz Eduardo Buzato. Composition of Meta-Objects in Guaraná. En *OOPSLA'98 Workshop on Reflective Programming in C++ and Java*, 1998. Updates IC-98-33.
- [OCGB98] Alexandre Oliva, Islene Calciolari Garcia y Luiz Eduardo Buzato. The Reflective Architecture of Guaraná. Technical Report IC-98-14, Instituto de Computação, Universidade Estadual de Campinas, Septiembre 1998.
- [OD01] Andrea Omicini y Enrico Denti. From Tuple Spaces to Tuple Centres. *Science of Computer Programming*, 41(3):277–294, Noviembre 2001.

- [OGT<sup>+</sup>99] Peyman Oreizy, Michael M. Gorlick, Richard N. Taylor, Dennis Heimbigner, Gregory Johnson, Nenad Medvidovic, Alex Quilici, David S. Rosenblum y Alexander L. Wolf. An Architecture-Based Approach to Self-Adaptive Software. *IEEE Intelligent Systems*, 14(3):54–62, Mayo 1999.
- [OIT93] Hideaki Okamura, Yutaka Ishikawa y Mario Tokoro. Metalevel Decomposition in AL-1/D. En Nishio y Yonezawa [NY93], págs. 110–127.
- [OK02] Harold Ossher y Gregor Kiczales, editores. *Proceedings of First International Conference on Aspect-Oriented Software Development (AOSD'02)*, Enschede, The Netherlands, Abril 2002. ACM Press.
- [OKH<sup>+</sup>95] Harold Ossher, Matthew Kaplan, William Harrison, Alexander Katz y Vincent Kruskal. Subject-Oriented Composition Rules. *ACM SIGPLAN Notices*, 30(10):235–250, Octubre 1995. Proceedings of OOPSLA'95.
- [Oli98] Alexandre Oliva. Guaraná: Uma Arquitetura de Software para Reflexão Computacional Implementada em Java. Tesis de Maestría, Universidade Estadual de Campinas, Septiembre 1998.
- [OMT98] Peyman Oreizy, Nenad Medvidovic y Richard N. Taylor. Architecture-Based Runtime Software Evolution. En *Proceedings of ICSE'98*, Kyoto, Japan, Abril 1998.
- [Ore96] Peyman Oreizy. Issues in the Runtime Modification of Software Architectures. Technical Report UCI-ICS-96-35, University of California Irvine, Agosto 1996.
- [OT98] Peyman Oreizy y Richard N. Taylor. On the Role of Software Architectures in Runtime System Reconfiguration. *IEE Proceedings-Software*, 145(8), Octubre 1998.
- [OT00] Harold Ossher y Peri Tarr. Multi-Dimensional Separation of Concerns and The Hyperspace Approach. En *Proceedings of the Symposium on Software Architectures and Component Technology: The State of the Art in Software Development*. Kluwer, 2000.
- [OZ99] Andrea Omicini y Franco Zambonelli. Coordination for Internet Application Development. *Journal of Autonomous Agents and Multi-Agent Systems*, 2(3):251–269, Septiembre 1999. Special Issue on Coordination Mechanisms for Web Agents.
- [PA97] George A. Papadopoulos y Farhad Arbab. Control-driven Coordination Programming in Shared Dataspace. En *Proceedings of PaCT'97*, volumen 1277 de *Lecture Notes in Computer Science*, págs. 247–261. Springer Verlag, 1997.
- [PA98a] George A. Papadopoulos y Farhad Arbab. Coordination Models and Languages. Technical Report SEN-R9834, Centrum voor Wiskunde en Informatica (CWI), Diciembre 1998.

- 
- [PA98b] Georgis A. Papadopoulos y Farhad Arbab. Coordination Models and Languages. En *The Engineering of Large Systems*, volumen 46 de *Advances in Computers*. Academic Press, Agosto 1998.
- [PA00] George A. Papadopoulos y Farhad Arbab. Dynamic Reconfiguration in Coordination Languages. En *Proceedings of High-Performance Computing and Networking 2000 (HPCN 2000)*, volumen 1823 de *Lecture Notes in Computer Science*, págs. 197–206, Amsterdam, 2000. Springer Verlag.
- [PA01] George A. Papadopoulos y Farhad Arbab. Configuration and Dynamic Reconfiguration of Components Using the Coordination Paradigm. *Future Generation Computer Systems*, 17(8):1023–1038, Junio 2001.
- [Par72] David Lorge Parnas. On the Criteria to Be Used in Decomposing Systems into Modules. *Communications of the ACM*, 15(12):1053–1058, Diciembre 1972.
- [Par01] Joachim Parrow. An Introduction to the  $\pi$ -calculus. En Bergstra et al. [BPS01], págs. 479–543.
- [PBC00] Jane Pryor, Natalio Bastán y Marcelo Campo. A Reflective Approach to Support Aspect Oriented Programming in Java. En Álvaro Ortigosa y Claudia A. Marcos, editores, *Proceedings of First Argentine Symposium on Software Engineering (ASSE 2000)*, en *Jornadas Argentinas de Informática e Investigación Operativa (29 JAIIO)*, Buenos Aires, Argentina, Septiembre 2000.
- [PD99] Norman W. Paton y Oscar Díaz. Active Database Systems. *ACM Computing Surveys*, 31(1):63–103, Marzo 1999.
- [Per85] Don Perlis. Languages with Self-reference I: Foundations (or: We can have everything in First-Order Logic!). *Artificial Intelligence*, 25:301–322, 1985.
- [Per88] Don Perlis. Commonsense Set Theory. En Maes y Nardi [MN88].
- [Per97] Dewayne E. Perry. State-of-the-Art: Software Architecture. En *Proceedings of the 19<sup>th</sup> International Conference on Software Engineering*, págs. 590–591. ACM Press, 1997.
- [Pie98] Benjamin C. Pierce. *Programming in the Pi-Calculus. A Tutorial Introduction to Pict*. Indiana University, Marzo 1998. (Pict Version 4.1).
- [Pit01] Andrew M. Pitts. Nominal Logic: A First Order Theory of Names and Binding. En Naoki Kobayashi y Benjamin C. Pierce, editores, *Fourth International Symposium on Theoretical Aspects of Computer Software (TACS 2001)*, volumen 2115 de *Lecture Notes in Computer Science*, págs. 219–242, Sendai, Japan, Octubre 2001. Springer-Verlag.
- [Plo75a] Gordon Plotkin. Call-by-name, call-by-value and the  $\lambda$ -calculus. *Theoretical Computer Science*, 1:125–159, 1975.

- [Plo75b] Gordon D. Plotkin. Call-by-Name, Call-by-Value and the  $\lambda$ -Calculus. *Theoretical Computer Science*, 1:125–159, 1975.
- [PR00] António Porto y Gruia-Catalin Roman, editores. *Fourth International Conference on Coordination Models and Languages (Coordination'00)*, volumen 1906 de *Lecture Notes in Computer Science*, Limassol, Cyprus, Septiembre 2000. Springer Verlag.
- [Pra86] Vaughan Pratt. Modelling Concurrency with Partial Orders. *International Journal of Parallel Programming*, 15:33–71, 1986.
- [Pra95] Vaughan R. Pratt. Chu Spaces and their Interpretation as Concurrent Objects. En J. van Leeuwen, editor, *Computer Science Today: Recent Trends and Developments*, volumen 1000 de *Lecture Notes in Computer Science*, págs. 392–405. Springer Verlag, 1995.
- [Pry00] Nathaniel G. Pryce. *Component Interaction in Distributed Systems*. Tesis Doctoral, Imperial College of Science, Technology and Medicine. University of London, Enero 2000.
- [PS96] Benjamin C. Pierce y Davide Sangiorgi. Typing and Subtyping for Mobile Processes. *Mathematical Structures in Computer Science*, 6(5), 1996.
- [PS98] John Peterson y Martin Sulzmann. Analysis of Architectures using Constraint-Based Types. Technical Report YALEU/DCS/RR-1157, Department of Computer Science, University of Yale, 1998.
- [PS99] John Peterson y Martin Sulzmann. Analysis of Architectures using Constraint-Based Types. En Clements et al. [CPR99].
- [PS00] Benjamin C. Pierce y Davide Sangiorgi. Behavioural Equivalence in the Polymorphic  $\pi$ -calculus. *Journal of the ACM*, 47(5):531–584, 2000.
- [PST00] Gordon Plotkin, Colin Stirling y Mads Tofte, editores. *Proof, Language, and Interaction: Essays in Honour of Robin Milner*. Foundations of Computing. MIT Press, Mayo 2000.
- [PT98] Benjamin C. Pierce y David N. Turner. *Pict Libraries Manual*. Indiana University, University of Glasgow, Marzo 1998. Pict 4.1.
- [PT00] Benjamin C. Pierce y David N. Turner. Pict: A Programming Language Based on the Pi-Calculus. En Plotkin et al. [PST00].
- [Pur90] James M. Purtilo. The POLYLITH Software Bus. Technical Report CS-TR-2469, University of Maryland, Institute for Advanced Computer Studies, Septiembre 1990.
- [Pur94] James M. Purtilo. The Polyolith Software Bus. *ACM Transactions on Programming Languages and Systems*, 16(1):151–174, 1994.

- 
- [PW92] Dewayne E. Perry y Alexander L. Wolf. Foundations for the Study of Software Architecture. *ACM Software Engineering Notes*, 17(4), Octubre 1992.
- [Qui85] Willard Van Orman Quine. Events and Reification. En E. Lepore y B. McLaughlin, editores, *Actions and Events*, págs. 162–171. Blackwell, 1985.
- [Qui92] Willard Van Orman Quine. *La Búsqueda de la Verdad*. Crítica, 1992. Trad. Javier Rodríguez Alcázar.
- [Ram25] Frank P. Ramsey. The Foundations of Mathematics. *Proceedings of the London Mathematical Society*, págs. 338–384, 1925.
- [Ran99] Alexander Ran. Software Isn't Built From Lego Blocks. En *Proceedings of ACM Symposium on Software Reusability (SSR'99)*, 1999.
- [RBP<sup>+</sup>91] James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy y William Lorenzen. *Object-Oriented Modeling and Design*. Prentice-Hall, 1991.
- [RC00] Barry Redmond y Vinny Cahill. Iguana/J: Towards a Dynamic and Efficient Reflective Architecture for Java. En Cazzola et al. [CCL00a].
- [RE96] Matthias Radestock y Susan Eisenbach. Coordination in Evolving Systems. En O. Spaniol, C. Linnhoff-Popien y B. Meyer, editores, *Trends in Distributed Systems - CORBA and Beyond*, volumen 1161 de *Lecture Notes in Computer Science*, págs. 162–176. Springer-Verlag, Berlin, 1996.
- [Rid99a] François-René Rideau. Métaprogrammation et Libre Disponibilité des Sources. Deux défis informatiques d'aujourd'hui. En *Autour du Libre 1999*, 1999.
- [Rid99b] François-René Rideau. Reflection, Non-Determinism, and the  $\lambda$ -Calculus. Rapport de recherche à paraître, CNET, 1999.
- [RMP97] Gruia-Catalin Roman, Peter J. McCann y Jerome Y. Plun. Mobile UNITY: Reasoning and Specification in Mobile Computing. *ACM Transactions on Software Engineering and Methodology*, 6(3):250–282, Julio 1997.
- [RRV98] António Ravara, Pedro Resende y Vasco T. Vasconcelos. Towards an Algebra of Dynamic Object Types. En *Workshop on Semantics of Objects as Processes*, número NS-98-5 en BRICS Notes Series, 1998.
- [RS96] Michael D. Rice y Stephen B. Seidman. Using Z as a Substrate for an Architectural Style Description Language. Technical Report CS-96-120, Department of Computer Science, Colorado State University, Septiembre 1996.
- [RT94] Jan Rutten y Daniele Turi. Initial Algebra and Final Coalgebra Semantics for Concurrency. En J.W. de Bakker, W.P. de Roever y G. Rozenberg, editores, *A Decade of Concurrency*, volumen 803 de *Lecture Notes in Computer Science*, págs. 530–582. Springer-Verlag, Berlín, 1994.

- [Rus08] Bertrand Russell. Mathematical Logic as Based on the Theory of Types. *American Journal of Mathematics*, 30:222–262, 1908.
- [Rus22] Bertrand Russell. Introduction. En *Ludwig Wittgenstein's Tractatus Logico-Philosophicus*. Keegan Paul, Trench & Tubner, Mayo 1922.
- [RV00] António Ravara y Vasco T. Vasconcelos. Typing Non-uniform Concurrent Objects. En *Proceedings of Concur'00*, volumen 1877 de *Lecture Notes in Computer Science*, págs. 474–488. Springer Verlag, 2000.
- [RW97] Anthony Rowstron y Alan Wood. Bonita: a set of tuple space primitives for distributed coordination. En *Proc. HICSS30, Sw Track*, págs. 379–388, Hawaii, 1997. IEEE Computer Society Press.
- [San92] Davide Sangiorgi. From  $\pi$ -calculus to Higher-Order  $\pi$ -calculus — and Back. En *Proceedings of TAPSOFT'93*, volumen 668 de *Lecture Notes in Computer Science*. Springer Verlag, 1992.
- [San93] Davide Sangiorgi. *Expressing Mobility in Process Algebra*. Tesis Doctoral, University of Edinburgh, 1993.
- [Sar97] Kamran Sartipi. A Survey on Software Architecture Domain. University of Waterloo, Ontario, Canadá, Febrero 1997.
- [Sch99] Jean-Guy Schneider. *Components, Scripts, and Glue: A conceptual Framework for Software Composition*. Tesis Doctoral, Institut für Informatik und angewandte Mathematik, Universität Bern, Octubre 1999.
- [SDK<sup>+</sup>95a] Mary Shaw, Robert DeLine, Daniel V. Klein, Theodore L. Ross, David M. Young y Gregory Zelesnik. Abstractions for Software Architecture and Tools to Support Them. *IEEE Transactions on Software Engineering*, 21(4):314–335, Abril 1995.
- [SDK<sup>+</sup>95b] Mary Shaw, Robert DeLine, Daniel V. Klein, Theodore L. Ross, David M. Young y Gregory Zelesnik. Abstractions for Software Architecture and Tools to Support Them. *IEEE Transactions on Software Engineering*, 21(4):314–335, Abril 1995.
- [Sew00] Marc Sewell. Are You A Software Architect? *Enterprise Development*, 2(5):62–63, Mayo 2000. Fawcette Technical Publications.
- [SG94] Mary Shaw y David Garlan. Characteristics of Higher-Level Languages for Software Architecture. Technical Report CMU-CS-94-210, SEI-94-TR-23, ESC-TR-94-023, Software Engineering Institute, Carnegie Mellon University, 1994.
- [SG95] Mary Shaw y David Garlan. Formulations and Formalisms in Software Architecture. En *Computer Science Today*, volumen 1000 de *Lecture Notes in Computer Science*. Springer-Verlag, 1995.
- [SG96] Mary Shaw y David Garlan. *Software Architecture: Perspectives on an Emerging Discipline*. Prentice-Hall, New Jersey, 1996.

- 
- [SG01] Bridget Spitznagel y David Garlan. A Compositional Approach for Constructing Connectors. En Kazman et al. [KKVvV01], págs. 148–157.
- [Sha80] Mary Shaw. The Impact of Abstraction Concerns on Modern Programming Languages. En *IEEE Proceedings*, Septiembre 1980. Reprinted: *IEEE Software*, Oct 1994.
- [Sha90] Mary Shaw. Prospects for an Engineering Discipline of Software. *IEEE Software*, Noviembre 1990. Reprints: SEI Technical Report CMU/SEI-90-TR-020. *Software Management*, D. Reifer (ed.), IEEE Press 1993. *Encyclopedia of Software Engineering*, Volume 2, pp. 930–940, John Wiley 1994.
- [Sha94] Mary Shaw. Procedure Calls Are the Assembly Language of Software Interconnection: Connectors Deserve First-Class Status. En *Proceedings of Workshop on Studies of Software Design*, Enero 1994.
- [Sha01] Mary Shaw. The Coming-of-Age of Software Architecture Research. En *Proceedings of the 23<sup>rd</sup> International Conference on Software Engineering (ICSE'01)*, Mayo 2001.
- [Sie92] Carles Sierra. Estructuras de Control en Inteligencia Artificial. Arquitecturas Reflexivas y de Meta Nivel. En *Nuevas Tendencias en Inteligencia Artificial*, págs. 91–110. Universidad de Deusto, 1992.
- [SK02] Marcelo Sihman y Shmuel Katz. A calculus of Superimpositions for Distributed Systems. En Ossher y Kiczales [OK02], págs. 28–40.
- [SL97] Jean-Guy Schneider y Markus Lumpe. Synchronizing Concurrent Objects in the  $\pi$ -Calculus. En Roland Ducournau y Serge Garlatti, editores, *Proceedings of Languages et Modèles à Objets*, págs. 61–76, Roscoff, Octubre 1997. Hermes.
- [Smi82] Brian Cantwell Smith. Reflection and Semantics in a Procedural Language. Technical Report MIT-LCS-TR-272, Massachusetts Institute of Technology, 1982.
- [Smo94] Gert Smolka. A Foundation for Higher-Order Concurrent Constraint Programming. En Jean-Pierre Jouannaud, editor, *Proceedings of the First International Conference of Constraints in Computational Logics*, volumen 845 de *Lecture Notes in Computer Science*, págs. 50–72. Springer Verlag, 1994.
- [SN99] Jean-Guy Schneider y Oscar M. Nierstrasz. Components, Scripts and Glue. En Leonor Barroca, Jon Hall y Patrick Hall, editores, *Software Architectures – Advances and Applications*, págs. 13–25. Springer Verlag, 1999.
- [SR98] Bran Selic y Jim Rumbaugh. Using UML for Modelling Complex Real-Time Systems. ObjectTime Ltd., Marzo 1998.
- [Ste94] Patrick Steyaert. *Open Design of Object-Oriented Languages: A Foundation For Specialisable Reflective Language Frameworks*. Tesis Doctoral, Vrije Universteit Brussel, 1994.

- [Sul94] Kevin J. Sullivan. *Mediators: Easing the Design and Evolution of Integrated Systems*. Tesis Doctoral, University of Washington, 1994. UW-CSE-TR-94-08-01.
- [SVK98] Sylvia Stuurman y Jan Van Katwijk. On-Line Change Mechanisms: The Software Architectural Level. En *6<sup>th</sup> International Symposium on the Foundations of Software Engineering*, 1998.
- [SW01] Davide Sangiorgi y David Walker. *The  $\pi$ -Calculus. A Theory of Mobile Processes*. Cambridge University Press, Junio 2001.
- [Szy98] Clemens Szyperski. *Component Software: Beyond Object-Oriented Programming*. Addison-Wesley, Enero 1998.
- [Tal96] Carolyn Talcott. An Actor Rewriting Theory. En *Proceedings of First International Workshop on Rewriting Logic and its Applications*, volumen 4 de *Electronic Notes in Theoretical Computer Science*, Septiembre 1996.
- [Tan91] Tomoyuki Tanaka. Actor Reflection without Meta-Objects. *OOPS Messenger*, 2(2), Abril 1991. ACM SIGPLAN.
- [Tar35] Alfred Tarski. The Concept of Truth in Formalized Languages. En *Logic, Semantics, Metamathematics*, págs. 152–278. Clarendon Press, 1935. Traducido por J. H. Woodger.
- [Tar44] Alfred Tarski. The Semantic Conception of Truth and the Foundations of Semantics. *Philosophical and Phenomenological Research*, 4, 1944.
- [Tho90] Bent Thomsen. *Calculi for Higher Order Communicating Systems*. Tesis Doctoral, Imperial College, University of London, Septiembre 1990.
- [TMA<sup>+</sup>96] Richard N. Taylor, Nenad Medvidovic, Kenneth M. Anderson, E. James Whitehead, Jr., Jason E. Robbins, Kari A. Nies, Peyman Oreizy y Deborah L. Dubrow. A Component- and Message-Based Architectural Style for GUI Software. *IEEE Transactions on Software Engineering*, 22(6):390–406, Junio 1996.
- [Tok90] Mario Tokoro. Computational Field Model: Toward a New Computing Model/Methodology for Open Distributed Environment. Technical Report SCSL-TR-90-006, Sony Computer Science Laboratory, 1990.
- [Tra96] Will Tracz. Minutes of the EDCS Lake Placid Kickoff Meeting. EDCS Community Pages, Julio 1996.
- [TRG02] Robert Tolksdorf y Gregor Rojec-Goldmann. The SPACETUB Models and Framework. En Arbab y Talcott [AT02], págs. 348–363.
- [Tur96] David N. Turner. *The Polymorphic Pi-Calculus: Theory and Implementation*. Tesis Doctoral, University of Edinburgh, 1996.

- [Ung88] David Ungar. Are Classes Obsolete? *ACM SIGPLAN Notices*, 23(12), 1988. OOPSLA'88.
- [Ung00] Victoria Ungureanu. *A Mechanism for Supporting Communication Policies in Distributed Systems*. Tesis Doctoral, Faculty of Computer Science, Rutgers University, 2000.
- [US87] David Ungar y Randall B. Smith. Self: The Power of Simplicity. En *ACM SIGPLAN Notices* [Mey87], págs. 227–242. OOPSLA'87.
- [Vas94a] Vasco T. Vasconcelos. Predicative Polymorphism in the  $\pi$ -calculus. En *Parallel Architectures and Languages Europe: PARLE'94*, volumen 817 de *Lecture Notes in Computer Science*, págs. 425–437. Springer Verlag, 1994.
- [Vas94b] Vasco Thudichum Vasconcelos. Typed Concurrent Objects. En Mario Tokoro y Remo Pareschi, editores, *ECOOOP'94: European Conference on Object-Oriented Programming*, volumen 821 de *Lecture Notes in Computer Science*, págs. 100–117. Springer Verlag, 1994.
- [Vas99] Vasco T. Vasconcelos. Processes, Functions, Datatypes. *Theory and Practice of Object Systems*, 5(2):97–110, 1999.
- [Vas01] Vasco Thudichum Vasconcelos. TyCO Gently. Technical Report DI-FCUL TR-01-4, Departamento de Informática, Universidade de Lisboa, Julio 2001.
- [Vay00] Julien Vayssière. Security and Meta Programming in Java. En Cazzola et al. [CCL00a].
- [vG01] Robert J. van Glabbeek. The Linear Time – Branching Time Spectrum I. The Semantics of Concrete, Sequential Processes. En Bergstra et al. [BPS01], págs. 3–99. Report CS-R9029, CWI, Amsterdam.
- [VH93] Vasco Thudichum Vasconcelos y Kohei Honda. Principal Typing Schemes in a Polyadic Pi-Calculus. En *Proceedings of 4<sup>th</sup> International Conference on Concurrency Theory (CONCUR'93)*, volumen 715 de *Lecture Notes in Computer Science*, págs. 524–538, 1993.
- [Vic95] Björn Victor. *The Mobility Workbench User's Guide. Polyadic version 3.122*. SICS, Estocolmo, Octubre 1995.
- [Vic98] Björn Victor. *The Fusion Calculus: Expressiveness and Symmetry in Mobile Processes*. Tesis Doctoral, Uppsala University, Sweden, Junio 1998.
- [VLS98] Vasco T. Vasconcelos, Luís Lopes y Fernando Silva. Distribution and Mobility with Lexical Scoping in Process Calculi. *Electronic Notes in Theoretical Computer Science*, 16(3), 1998. Third International Workshop on High-Level Concurrent Languages.

- [VM94] Björn Victor y Faron Moller. The Mobility Workbench — A Tool for the  $\pi$ -Calculus. En David Dill, editor, *CAV'94: Computer Aided Verification*, volumen 818 de *Lecture Notes in Computer Science*, págs. 428–440. Springer-Verlag, 1994.
- [VM99] Antonio Vallecillo Moreno. *Un Modelo de Componentes para el Desarrollo de Aplicaciones Distribuidas*. Tesis Doctoral, Universidad de Málaga, Marzo 1999.
- [vOvdLKM00] Rob van Ommering, Frank van der Linden, Jeff Kramer y Jeff Magee. The Koala Component Model for Consumer Electronics Software. *IEEE Computer*, 33(3):78–85, 2000.
- [VPL99] James Vera, Louis Perrochon y David C. Luckham. Event-Based Execution Architectures for Dynamic Software Systems. En Donohoe [Don99], págs. 303–318.
- [vR01] Guido van Rossum. *Python Reference Manual, release 1.5.2*. Stichting Mathematisch Centrum, Amsterdam, Marzo 2001.
- [VRMP01] Federico Valentino, Andrés Ramos, Claudia A. Marcos y Jane Pryor. A Framework for the Development of Multi-Level Reflective Applications. En *Proceedings of Second Argentine Symposium on Software Engineering (ASSE 2001)*, Buenos Aires, Argentina, Septiembre 2001.
- [VT93] Vasco Thudichum Vasconcelos y Mario Tokoro. A Typing System for a Calculus of Objects. En Nishio y Yonezawa [NY93], págs. 460–474.
- [VT01] Nalini Venkatasubramanian y Carolyn Talcott. A Semantic Framework for Reasoning about Reflective Middleware. *IEEE Distributed Systems Online*, 2(6), 2001.
- [Wal95] David Walker. Objects in the  $\pi$ -Calculus. *Journal of Information and Computation*, 116(2):253–271, 1995.
- [Wer98] Michel Wermelinger. Towards a Chemical Model for Software Architecture Reconfiguration. *IEE Proceedings – Software*, 145(5):130–136, Octubre 1998.
- [Wer99] Michel Alexandre Wermelinger. *Specification of Software Architecture Reconfiguration*. Tesis Doctoral, Universidade Nova de Lisboa, Septiembre 1999.
- [WF88] Mitchell Wand y Daniel P. Friedman. The Mystery of the Tower Revealed: A Nonreflective Description of the Reflective Tower. *Lisp and Symbolic Computation*, 1, 1988.
- [WF99] Michel Wermelinger y José Luiz Fiadeiro. Algebraic Software Architecture Reconfiguration. En *Software Engineering – Proceedings of ESEC/FSE'99*, volumen 1687 de *Lecture Notes in Computer Science*, págs. 393–409. Springer Verlag, 1999.

- 
- [Whi97] Scott A. Whitmire. *Object-Oriented Design Measurement*. John Wiley & Sons, Inc., 1997.
- [Wil99] David Wile. AML: an Architecture Meta-Language. En *Proceedings of ASE'99*, págs. 183–190. Octubre, IEEE-CS Press, 1999.
- [WK99] Jos Warmer y Anneke Kleppe. *The Object Constraint Language*. Object Technology Series. Addison-Wesley, 1999.
- [WLF00] Michel Wermelinger, Antónia Lopes y José Luiz Fiadeiro. Superposing Connectors. En *Proceedings of 10<sup>th</sup> International Workshop on Software Specification and Design*, págs. 87–94. IEEE Computer Society Press, 2000.
- [Woj00] Paweł T. Wojciechowski. The Nomadic Pict System. Documentation and User's Manual. Release 1.0a. Technical report, University of Cambridge, Computer Laboratory, Diciembre 2000.
- [Wol97] Alexander L. Wolf. Succeedings of the Second International Software Architecture Workshop (ISAW-2). *ACM Software Engineering Notes*, 22(1):42–56, Enero 1997. ACM SIGSOFT'96, FSE-4.
- [WS98] Ian S. Welch y Robert J. Stroud. Adaptation of Connectors in Software Architectures. En Wolfgang Weck, Jan Bosch y Clemens Szyperski, editores, *Proceedings of the Third International Workshop on Component-Oriented Programming (WCOP'98)*, volumen 10 de *TUCS General Publications*. Turku Centre for Computer Science (TUCS), Octubre 1998.
- [WS00] Paweł T. Wojciechowski y Peter Sewell. Nomadic Pict: Language and Infrastructure Design for Mobile Agents. *IEEE Concurrency*, 8(2):42–52, 2000.
- [WY88] Takuo Watanabe y Akinori Yonezawa. Reflection in an Object-Oriented Concurrent Language. En *OOPSLA '88 Proceedings*, págs. 305–315, Septiembre 1988.
- [YM01] Akinori Yonezawa y Satoshi Matsuoka, editores. *Reflection 2001: Third International Conference on Metalevel Architectures and Separation of Crosscutting Concerns*, volumen 2192 de *Lecture Notes in Computer Science*, Kyoto, Japan, Septiembre 2001. AITO, ACM SIGSOFT, JSST, Springer Verlag.
- [Yok92] Yasuhiko Yokote. The Apertos Reflective Operating System: The Concept and Its Implementation. Technical Report SCLS-TR-92-014, Sony Computer Science Laboratory Inc., Tokyo, Japan, Junio 1992.
- [YT87] Yasuhiko Yokote y Mario Tokoro. Experience and Evolution of ConcurrentS-malltalk. En Meyrowitz [Mey87], págs. 406–415.
- [Zav98] Gianluigi Zavattaro. On the Incomparability of Gamma and Linda. Technical Report SEN-R9827, Centrum voor Wiskunde en Informatica (CWI), Octubre 1998.

## Bibliografía

---

- [ZP97] Xiaogang Zhang y John Potter. Class-based Models in the  $\pi$ -Calculus. En Christine Mingins, Roger Duke y Bertrand Meyer, editores, *Proceedings of the 25<sup>th</sup> International Conference in Technology of Object-Oriented Languages and Systems (TOOLS Pacific '97)*, págs. 219–231, Melbourne, Australia, Noviembre 1997.
- [ZP98] Xiaogang Zhang y John Potter. A Compositional Approach to Concurrent Objects. En Jian Chen, M. Li, Christine Mingis y Bertrand Meyer, editores, *27<sup>th</sup> Conference on Technology of Object-Oriented Languages and Systems (TOOLS Asia '98)*, Beijing, China, Septiembre 1998. IEEE Computer Society Press.
- [ZW95] Amy Moormann Zaremski y Jeannette Wing. Specification Matching of Software Components. En *Proceedings of the ACM SIGSOFT'95 Foundations of Software Engineering Symposium*. ACM Press, 1995.

